aws

Guide du développeur

Amazon Comprehend

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Comprehend: Guide du développeur

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques commerciales et la présentation commerciale d’Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d’une manière susceptible d’entraîner une confusion chez les clients, ou d’une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce qu'Amazon Comprehend ? 1
Informations sur Amazon Comprehend 2
Amazon Comprehend personnalisé 2
Volants 3
Regroupement de documents (modélisation thématique) 3
Exemples 3
Avantages 4
Tarification d'Amazon Comprehend 5
Utilisez-vous Amazon Comprehend pour la première fois ? 5
Comment ça marche 7
Informations 7
Entités 8
Événements 10
Phrases clés 18
Langue dominante 19
Sentiment 26
Sentiment ciblé 27
Analyse syntaxique 44
Amazon Comprehend personnalisé 48
Modélisation des rubriques 49
Modes de traitement des documents 53
Traitement d'un seul document 54
Traitement synchrone de plusieurs documents 54
Traitement par lots asynchrone 57
Langues prises en charge 58
Langues prises en charge 58
Langues prises en charge par les fonctionnalités d'Amazon Comprehend 59
Configuration 61
Inscrivez-vous pour un Compte AWS 61
Création d’un utilisateur doté d’un accès administratif 62
Configurez le AWS CLI 63
Octroi d’un accès par programmation 64
Premiers pas 66
Utilisation de la console 68

Analyse en temps réel ... 68
Entités ... 69
Phrases clés ... 70
Langue ... 71
données d'identification personnelle (PII) ... 72
Sentiment ... 74
Sentiment ciblé ... 75
Syntaxe ... 77
Tâches d'analyse (console) ... 78
Utilisation de l’API ... 82
Travailler avec AWS SDKs ... 82
Analyse en temps réel (API) ... 83
Détecter la langue dominante ... 84
Détection des entités nommées ... 85
Détecter les phrases clés ... 86
Déterminer le sentiment ... 87
Analyse en temps réel pour un sentiment ciblé ... 89
Détection de la syntaxe ... 91
Lot en temps réel APIs ... 94
Tâches d'analyse asynchrones (API) ... 99
Informations sur Amazon Comprehend ... 99
Sentiment ciblé ... 105
Détection d'événements ... 107
Modélisation des rubriques ... 112
Confiance et sécurité ... 116
Détection de toxicité ... 117
Détection du contenu toxique à l'aide de l'API ... 118
Classification de sécurité rapide ... 120
Classification de sécurité rapide à l'aide de l'API ... 121
Détection et rédaction des informations personnelles ... 123
données d'identification personnelle (PII) ... 124
Détection des entités PII ... 124
Localiser les entités PII ... 125
Rédiger les entités PII ... 126
Types d'entités universelles PII ... 126
Types d'entités PII spécifiques à chaque pays ... 129

Étiquetage des entités PII 131
Analyse en temps réel (console) 132
Offsets 72
Étiquettes 74
Tâches d'analyse asynchrones (console) 135
Analyse en temps réel (API) 137
Localisation des entités PII en temps réel (API) 138
Étiquetage des entités PII en temps réel (API) 139
Tâches d'analyse asynchrones (API) 140
Localisation des entités PII 140
Rédaction d'entités PII 145
Traitement des documents 150
Entrées pour une analyse en temps réel 150
Documents en texte brut 151
Documents semi-structurés 151
Fichiers image et fichiers PDF numérisés 151
Sortie Amazon Textract 151
Tailles de document maximales pour une analyse en temps réel 151
Erreurs dans les documents semi-structurés 152
Entrées pour l'analyse asynchrone 153
Documents en texte brut 153
Documents semi-structurés 154
Fichiers image et fichiers PDF numérisés 155
Fichiers JSON de sortie Amazon Textract 155
Configuration des options d'extraction de texte 155
Bonnes pratiques pour les images 157
Classification personnalisée 158
Préparation des données d'entraînement 159
Formats de fichiers de formation 160
Mode multi-classes 161
Mode multi-étiquettes 164
Modèles de classification des formations 168
Entraînez des classificateurs personnalisés (console) 169
Entraînez des classificateurs personnalisés (API) 174
Testez les données d'entraînement 176
Résultat d'entraînement du classificateur 177

Métriques 182
Exécution d'analyses en temps réel 187
Analyse en temps réel (console) 188
Analyse en temps réel (API) 190
Sorties pour une analyse en temps réel 193
Exécution de tâches d'analyse asynchrones 195
Formats de fichiers d'entrée 196
Tâches d'analyse (console) 197
Tâches d'analyse (API) 199
Sorties pour les tâches d'analyse 201
Reconnaissance d'entités personnalisée 206
Préparation des données d'entraînement 207
Quand utiliser des annotations plutôt que des listes d'entités 208
Listes d'entités 209
Annotations 212
Modèles de reconnaissance d'entraînement 226
Entraînez des outils de reconnaissance personnalisés (console) 227
Formez des outils de reconnaissance personnalisés (API) 234
Métriques 237
Exécution d'analyses en temps réel 241
Analyse en temps réel (console) 242
Analyse en temps réel (API) 244
Sorties pour une analyse en temps réel 246
Exécution de tâches d'analyse asynchrones 253
Tâches d'analyse (console) 254
Tâches d'analyse (API) 256
Sorties pour les tâches d'analyse 259
Gestion de modèles personnalisés 265
Versionnage de modèles avec Amazon Comprehend 265
Copier des modèles personnalisés entre Comptes AWS 268
Partage d'un modèle personnalisé 269
Importation d'un modèle personnalisé 279
Volants 286
Vue d'ensemble du volant 286
Ensembles de données Flywheel 287
Création d'un volant 288

États du volant 289
Itérations du volant 289
Lacs de données Flywheel 290
Structure des dossiers du lac de données 290
Gestion des lacs de données 291
Autorisations et politiques IAM 292
Configuration des autorisations utilisateur IAM 292
Configurer les autorisations pour les AWS KMS clés 293
Création d'un rôle d'accès aux données 293
Configuration des volants (console) 293
Créez un volant 294
Mettre à jour un volant 296
Supprimer un volant 297
Configuration des volants (API) 297
Création d'un volant pour un modèle existant 298
Création d'un volant pour un nouveau modèle 298
Décrire un volant 299
Mettre à jour un volant 300
Supprimer un volant 300
Listez les volants 301
Configuration des ensembles de données 301
Création d'un jeu de données (console) 302
Création d'un jeu de données (API) 302
Décrire un ensemble de données 303
Itérations du volant 303
Flux de travail d'itération 304
Gestion des itérations (console) 304
Gestion des itérations (API) 305
Utilisation de volants 308
Analyse en temps réel 308
Tâches asynchrones 309
Gestion des points de terminaison 310
Vue d'ensemble des endpoints 310
Utilisation des points de terminaison 311
Surveillance des points de terminaison 312
Mise à jour des terminaux 315

En utilisant Trusted Advisor 317
Points de terminaison sous-utilisés d'Amazon Comprehend 317
Risque lié à l'accès aux terminaux Amazon Comprehend 320
Suppression de points de terminaison 322
Mise à l'échelle automatique avec les points de terminaison 323
Suivi de la cible 323
Mise à l’échelle planifiée 328
Identification 333
Marquer une nouvelle ressource 334
Afficher, modifier et supprimer des balises 335
Exemples de code 337
Principes de base 338
Actions 339
Scénarios 411
Créer une application de streaming Amazon Transcribe 412
Création d'un chatbot Amazon Lex 412
Création d'une application de messagerie 414
Créez une application pour analyser les commentaires des clients 415
Détecter les éléments du document 421
Détecter des entités dans un texte extrait à partir d'une image 427
Exécuter une tâche de modélisation de rubrique sur des exemples de données 427
Formez un classificateur personnalisé et classez des documents 432
Sécurité 446
Protection des données 447
Chiffrement KMS dans Amazon Comprehend 448
Prévention du problème de l’adjoint confus entre services 451
Utilisation d'un Virtual Private Cloud (VPC) 454
Points de terminaison d’un VPC (AWS PrivateLink) 461
Gestion de l’identité et des accès 463
Public ciblé 463
Authentification par des identités 464
Gestion des accès à l’aide de politiques 468
Comment Amazon Comprehend fonctionne avec IAM 471
Exemples de politiques basées sur l’identité 479
AWS politiques gérées 492
Résolution des problèmes 496

Journalisation des appels d'API Amazon Comprehend avec AWS CloudTrail 498
Informations sur Amazon Comprehend dans CloudTrail 498
Exemple : entrées du fichier journal Amazon Comprehend 501
Validation de conformité 503
Résilience 503
Sécurité de l'infrastructure 504
Consignes et quotas 505
Régions prises en charge 505
Quotas pour les modèles intégrés 506
Analyse en temps réel (synchrone) 506
Analyse asynchrone 508
Quotas pour les modèles personnalisés 511
Quotas généraux 511
Quotas pour les terminaux 512
Classification des documents 512
Reconnaissance d'entités personnalisée 517
Quotas pour les volants 521
Quotas généraux pour les volants 521
Quotas de jeux de données pour les modèles de classification personnalisés 522
Quotas de jeux de données pour les modèles de reconnaissance d'entités personnalisés ... 522
Didacticiels 524
Analyse des informations issues des avis 524
Prérequis 526
Étape 1 : ajout de documents à Amazon S3 528
Étape 2 : (CLI uniquement) création d'un rôle IAM 532
Étape 3 : Exécution de tâches d'analyse 536
Étape 4 : Préparation de la sortie 540
Étape 5 : visualisation de la sortie 553
Utilisation des points d'accès Lambda des objets S3 pour les informations personnelles 559
Contrôle de l'accès aux documents à l'aide des informations personnelles 560
Supprimer des informations personnelles à partir de documents 562
Analyser du texte avec OpenSearch 564
Référence d'API 565
Historique de la documentation 566
AWS Glossaire 584
..... dlxxxv

Qu'est-ce qu'Amazon Comprehend ?

Amazon Comprehend utilise le traitement du langage naturel (NLP) pour extraire des informations sur le contenu des documents. Il développe des informations en reconnaissant les identités, les phrases clés, le langage, les sentiments et d'autres éléments courants dans un document. Utilisez Amazon Comprehend pour créer de nouveaux produits basés sur la compréhension de la structure des documents. Par exemple, Amazon Comprehend vous permet de rechercher des mentions de produits sur les réseaux sociaux ou de scanner l'intégralité d'un référentiel de documents à la recherche de phrases clés.

Vous pouvez accéder aux fonctionnalités d'analyse de documents d'Amazon Comprehend à l'aide de la console Amazon Comprehend ou d'Amazon Comprehend. APIs Vous pouvez exécuter une analyse en temps réel pour de petites charges de travail ou démarrer des tâches d'analyse asynchrones pour de grands ensembles de documents. Vous pouvez utiliser les modèles préentraînés fournis par Amazon Comprehend, ou vous pouvez créer vos propres modèles personnalisés pour la classification et la reconnaissance d'entités.

Amazon Comprehend peut stocker votre contenu afin d'améliorer en permanence la qualité de ses modèles préentraînés. Consultez la FAQ Amazon Comprehend pour en savoir plus.

Toutes les fonctionnalités d'Amazon Comprehend acceptent les documents texte UTF-8 comme entrée. En outre, la classification personnalisée et la reconnaissance d'entités personnalisées acceptent les fichiers image, les fichiers PDF et les fichiers Word en entrée.

Amazon Comprehend peut examiner et analyser des documents dans différentes langues, en fonction de la fonctionnalité spécifique. Pour de plus amples informations, veuillez consulter Langues prises en charge dans Amazon Comprehend. La Langue dominante capacité d'Amazon Comprehend permet d'examiner des documents et de déterminer la langue dominante pour une sélection beaucoup plus large de langues.

Rubriques

- Informations sur Amazon Comprehend
- Amazon Comprehend personnalisé
- Volants
- Regroupement de documents (modélisation thématique)
- Exemples
- Avantages

- Tarification d'Amazon Comprehend
- Utilisez-vous Amazon Comprehend pour la première fois ?

Informations sur Amazon Comprehend

Amazon Comprehend utilise un modèle préformé pour examiner et analyser un document ou un ensemble de documents afin de recueillir des informations à son sujet. Ce modèle est continuellement entraîné sur un gros corps de texte, de sorte que vous n'avez pas besoin de fournir de données d'entraînement.

Amazon Comprehend analyse les types d'informations suivants :

- Entités : références aux noms de personnes, de lieux, d'objets et de lieux contenus dans un document.
- Phrases clés : phrases qui apparaissent dans un document. Par exemple, un document concernant un match de basket peut renvoyer les noms des équipes, le nom du site et le score final.
- Informations personnelles identifiables (PII) : données personnelles permettant d'identifier une personne, telles qu'une adresse, un numéro de compte bancaire ou un numéro de téléphone.
- Langue : langue dominante d'un document.
- Sentiment : sentiment dominant d'un document, qui peut être positif, neutre, négatif ou mitigé.
- Sentiment ciblé : sentiments associés à des entités spécifiques dans un document. Le sentiment associé à chaque occurrence d'entité peut être positif, négatif, neutre ou mixte.
- Syntaxe : parties du discours correspondant à chaque mot du document.

Pour de plus amples informations, veuillez consulter Informations.

Amazon Comprehend personnalisé

Vous pouvez personnaliser Amazon Comprehend en fonction de vos besoins spécifiques sans avoir les compétences requises pour créer des solutions NLP basées sur le machine learning. À l'aide de l'apprentissage automatique, ou AutoML, Amazon Comprehend Custom crée des modèles NLP personnalisés en votre nom, en utilisant les données dont vous disposez déjà.

Classification personnalisée — Créez des modèles de classification personnalisés (classificateurs) pour organiser vos documents dans vos propres catégories.

Reconnaissance d'entités personnalisée : créez des modèles de reconnaissance d'entités personnalisés (outils de reconnaissance) capables d'analyser le texte pour détecter des termes spécifiques et des phrases basées sur des noms.

Pour de plus amples informations, veuillez consulter Amazon Comprehend personnalisé.

Volants

Utilisez des volants pour simplifier le processus de formation et de gestion des versions de modèles personnalisées au fil du temps. Un volant permet d'orchestrer les tâches associées à la formation et à l'évaluation des nouvelles versions d'un modèle. Les volants prennent en charge les modèles personnalisés en texte brut pour une classification personnalisée et une reconnaissance personnalisée des entités. Pour de plus amples informations, veuillez consulter Volants.

Regroupement de documents (modélisation thématique)

Vous pouvez également utiliser Amazon Comprehend pour examiner un corpus de documents afin de les organiser en fonction de mots clés similaires qu'ils contiennent. Le regroupement de documents (modélisation de sujets) est utile pour organiser un vaste corpus de documents en sujets ou en groupes similaires en fonction de la fréquence des mots. Pour de plus amples informations, veuillez consulter Modélisation des rubriques.

Exemples

Les exemples suivants montrent comment vous pouvez utiliser les opérations Amazon Comprehend dans vos applications.

Example 1 : Rechercher des documents sur un sujet

Recherchez les documents relatifs à un sujet particulier à l'aide de la modélisation thématique Amazon Comprehend. Scannez un ensemble de documents pour déterminer les sujets abordés et pour trouver les documents associés à chaque sujet. Vous pouvez spécifier le nombre de sujets qu'Amazon Comprehend doit renvoyer à partir de l'ensemble de documents.

Example 2 : Découvrez ce que les clients pensent de vos produits

Si votre entreprise publie un catalogue, laissez Amazon Comprehend vous expliquer ce que les clients pensent de vos produits. Envoyez un commentaire à chaque client sur

l'`DetectSentiment`opération et celui-ci vous indiquera si les clients ont une opinion positive, négative, neutre ou mitigée à propos d'un produit.

Example 3 : Découvrez ce qui compte pour vos clients

Utilisez la modélisation de sujets Amazon Comprehend pour découvrir les sujets abordés par vos clients sur vos forums et forums, puis utilisez la détection d'entités pour déterminer les personnes, les lieux et les éléments qu'ils associent au sujet. Utilisez l'analyse des sentiments pour déterminer ce que vos clients pensent d'un sujet.

Avantages

Les avantages liés à l'utilisation d'Amazon Comprehend incluent :

- Intégrez un puissant traitement du langage naturel à vos applications : Amazon Comprehend simplifie la création de fonctionnalités d'analyse de texte dans vos applications en proposant un traitement du langage naturel puissant et précis via une API simple. Vous n'avez pas besoin d'expertise en analyse textuelle pour tirer parti des informations produites par Amazon Comprehend.
- Traitement du langage naturel basé sur le deep learning : Amazon Comprehend utilise la technologie d'apprentissage profond pour analyser le texte avec précision. Nos modèles sont constamment entraînés avec de nouvelles données dans plusieurs domaines afin d'améliorer la précision.
- Traitement évolutif du langage naturel : Amazon Comprehend vous permet d'analyser des millions de documents afin de découvrir les informations qu'ils contiennent.
- Intégré à d'autres AWS services, Amazon Comprehend est conçu pour fonctionner parfaitement avec d'autres AWS services tels qu'Amazon S3 et AWS KMS. AWS Lambda Stockez vos documents dans Amazon S3 ou analysez les données en temps réel avec Firehose. Support for AWS Identity and Access Management (IAM) permet de contrôler facilement et en toute sécurité l'accès aux opérations Amazon Comprehend. À l'aide d'IAM, vous pouvez créer et gérer des utilisateurs et des groupes afin d'accorder l'accès approprié à vos développeurs et utilisateurs finaux.
- Chiffrement des résultats de sortie et des volumes de données : Amazon S3 vous permet déjà de chiffrer vos documents d'entrée, et Amazon Comprehend va encore plus loin. À l'aide de votre propre clé KMS, vous pouvez chiffrer les résultats de sortie de votre tâche et les données du volume de stockage attaché à l'instance de calcul qui traite la tâche d'analyse. Il en résulte une sécurité nettement améliorée.

- Faible coût — Avec Amazon Comprehend, il n'y a pas de frais minimum ni d'engagement initial. Vous payez pour les documents que vous analysez et les modèles personnalisés que vous entraînez.

Tarification d'Amazon Comprehend

Avec Amazon Comprehend, vous ne payez que pour les ressources que vous utilisez. Si vous êtes un nouveau AWS client, vous pouvez commencer à utiliser Amazon Comprehend gratuitement. Pour plus d'informations, consultez le niveau d'utilisation AWS gratuit.

Des frais d'utilisation sont facturés pour l'exécution de tâches d'analyse en temps réel ou asynchrones. Vous payez pour former des modèles personnalisés et vous payez pour une gestion personnalisée des modèles. Pour les demandes en temps réel utilisant des modèles personnalisés, vous payez pour le point de terminaison à partir du moment où vous le démarrez jusqu'à ce que vous le supprimiez. Il n'y a pas de frais supplémentaires pour l'utilisation des volants. Toutefois, lorsque vous exécutez une itération en volant, vous devez payer les frais standard pour l'entraînement d'une nouvelle version du modèle et le stockage des données du modèle.

Pour connaître les tarifs et obtenir des informations détaillées supplémentaires, consultez Amazon Comprehend Pricing.

Utilisez-vous Amazon Comprehend pour la première fois ?

Si vous utilisez Amazon Comprehend pour la première fois, nous vous recommandons de lire les sections suivantes dans l'ordre :

1. Fonctionnement— Cette section présente les concepts d'Amazon Comprehend.
2. Configuration— Dans cette section, vous créez un compte et configurez le AWS CLI.
3. Commencer à utiliser Amazon Comprehend— Dans cette section, vous exécutez une tâche d'analyse Amazon Comprehend.
4. Tutoriel : Analyse des informations issues des avis clients avec Amazon Comprehend— Dans cette section, vous allez effectuer une analyse des sentiments et des entités et visualiser les résultats.
5. Amazon Comprehend API Reference : documentation de référence pour les opérations Amazon Comprehend.

AWS fournit les ressources suivantes pour en savoir plus sur le service Amazon Comprehend :

- Le blog AWS Machine Learning contient des articles utiles sur Amazon Comprehend.
- Amazon Comprehend Resources propose des vidéos et des didacticiels utiles sur Amazon Comprehend.

Fonctionnement

Amazon Comprehend utilise un modèle préformé pour recueillir des informations sur un document ou un ensemble de documents. Ce modèle est continuellement entraîné sur un gros corps de texte afin que vous n'ayez pas besoin de fournir de données d'entraînement.

Vous pouvez utiliser Amazon Comprehend pour créer vos propres modèles personnalisés pour une classification personnalisée et une reconnaissance d'entité personnalisée. Vous pouvez l'utiliser Volants pour aider à gérer les modèles personnalisés.

Amazon Comprehend fournit une modélisation thématique à l'aide d'un modèle intégré. La modélisation thématique examine un corpus de documents et organise les documents en fonction de mots clés similaires qu'ils contiennent.

Amazon Comprehend propose des modes de traitement de documents synchrones et asynchrones. Utilisez le mode synchrone pour traiter un document ou un lot de 25 documents maximum. Utilisez une tâche asynchrone pour traiter un grand nombre de documents.

Amazon Comprehend travaille avec AWS Key Management Service (AWS KMS) pour fournir un chiffrement amélioré de vos données. Pour de plus amples informations, veuillez consulter Chiffrement KMS dans Amazon Comprehend.

Concepts clés

- Informations
- Amazon Comprehend personnalisé
- Modélisation des rubriques
- Modes de traitement des documents

Informations

Amazon Comprehend peut analyser un document ou un ensemble de documents afin de recueillir des informations à son sujet. Parmi les informations développées par Amazon Comprehend à propos d'un document, citons :

- Entités— Amazon Comprehend renvoie une liste d'entités, telles que des personnes, des lieux et des lieux, identifiées dans un document.

- Événements— Amazon Comprehend détecte des types spécifiques d'événements et des informations connexes.
- Phrases clés— Amazon Comprehend extrait les phrases clés qui apparaissent dans un document. Par exemple, un document concernant un match de basket peut renvoyer les noms des équipes, le nom du site et le score final.
- données d'identification personnelle (PII) — Amazon Comprehend analyse les documents afin de détecter les données personnelles permettant d'identifier une personne, telles qu'une adresse, un numéro de compte bancaire ou un numéro de téléphone.
- Langue dominante— Amazon Comprehend identifie la langue dominante d'un document. Amazon Comprehend peut identifier 100 langues.
- Sentiment — Amazon Comprehend détermine le sentiment dominant d'un document. Le sentiment peut être positif, neutre, négatif ou mixte.
- Sentiment ciblé — Amazon Comprehend détermine le sentiment des entités spécifiques mentionnées dans un document. Le sentiment associé à chaque mention peut être positif, neutre, négatif ou mitigé.
- Analyse syntaxique— Amazon Comprehend analyse chaque mot de votre document et détermine la partie du discours correspondant à ce mot. Par exemple, dans la phrase « Il pleut aujourd'hui à Seattle », « il » est identifié comme un pronom, « il pleut » est identifié comme un verbe et « Seattle » est identifié comme un nom propre.

Entités

Une entité est une référence textuelle au nom unique d'un objet du monde réel tel que des personnes, des lieux et des articles commerciaux, et à des références précises à des mesures telles que les dates et les quantités.

Par exemple, dans le texte « John a déménagé au 1313 Mockingbird Lane en 2012 », « John » peut être reconnu comme un`PERSON`, « 1313 Mockingbird Lane » peut être reconnu comme un et « 2012 » peut être reconnu comme un`LOCATION`. `DATE`

Chaque entité possède également un score qui indique le niveau de confiance d'Amazon Comprehend quant à la détection correcte du type d'entité. Vous pouvez filtrer les entités présentant les scores les plus faibles afin de réduire le risque d'utiliser des détections incorrectes.

Le tableau suivant répertorie les types d'entités.

Type	Description
ARTICLE_COMMERCIAL	Un produit de marque
DATE	Une date complète (par exemple, le 25/11/2017), un jour (mardi), un mois (mai) ou une heure (8 h 30)
ÉVÉNEMENT	Un événement, tel qu'un festival, un concert, une élection, etc.
LOCATION	Un lieu spécifique, tel qu'un pays, une ville, un lac, un bâtiment, etc.
ORGANISATION	Les grandes organisations, telles qu'un gouvernement, une entrepris e, une religion, une équipe sportive, etc.
OTHER	Entités qui ne correspondent à aucune des autres catégories d'entités
LA PERSONNE	Individus, groupes de personnes, surnoms, personnages fictifs
QUANTITÉ	Un montant quantifié, tel que la devise, les pourcentages, les nombres, les octets, etc.
TITLE	Nom officiel donné à toute création ou œuvre créative, telle que des films, des livres, des chansons, etc.

Les opérations de détection des entités peuvent être effectuées dans l'un des principaux langages pris en charge par Amazon Comprehend. Cela inclut uniquement la détection d'entités prédéfinies (non personnalisées). Tous les documents doivent être rédigés dans la même langue.

Vous pouvez utiliser l'une des opérations d'API suivantes pour détecter des entités dans un document ou un ensemble de documents.

- DetectEntities
- BatchDetectEntities
- StartEntitiesDetectionJob

Les opérations renvoient une liste d'objets API Entity, un pour chaque entité du document. L'`BatchDetectEntities`opération renvoie une liste d'`Entity`objets, une liste pour chaque

document du lot. L'`StartEntitiesDetectionJob`opération démarre une tâche asynchrone qui produit un fichier contenant une liste d'`Entity`objets pour chaque document de la tâche.

L'exemple suivant est la réponse de l'`DetectEntities`opération.

```
{
    "Entities": [
        {
            "Text": "today",
            "Score": 0.97,
            "Type": "DATE",
            "BeginOffset": 14,
            "EndOffset": 19
        },
        {
            "Text": "Seattle",
            "Score": 0.95,
            "Type": "LOCATION",
            "BeginOffset": 23,
            "EndOffset": 30
        }
    ],
    "LanguageCode": "en"
}
```

Événements

Utilisez la détection d'événements pour analyser des documents texte afin de détecter des types d'événements spécifiques et leurs entités associées. Amazon Comprehend prend en charge la détection d'événements dans de grandes collections de documents à l'aide de tâches d'analyse asynchrones. Pour plus d'informations sur les événements, y compris des exemples de tâches d'analyse d'événements, consultez Annonce du lancement d'Amazon Comprehend Events

Entités

À partir du texte saisi, Amazon Comprehend extrait une liste d'entités associées à l'événement détecté. Une entité peut être un objet du monde réel, tel qu'une personne, un lieu ou un lieu ; une entité peut également être un concept, tel qu'une mesure, une date ou une quantité. Chaque occurrence d'une entité est identifiée par une mention, qui est une référence textuelle à l'entité dans le texte saisi. Pour chaque entité unique, toutes les mentions sont regroupées dans une liste. Cette

liste fournit des détails pour chaque emplacement dans le texte d'entrée où l'entité apparaît. Amazon Comprehend détecte uniquement les entités associées aux types d'événements pris en charge.

Chaque entité associée à un type d'événement pris en charge renvoie les informations associées suivantes :

- Mentions : détails pour chaque occurrence de la même entité dans le texte saisi.
  - BeginOffset: décalage de caractères dans le texte saisi qui indique le début de la mention (le premier caractère est à la position 0).
  - EndOffset: décalage de caractères dans le texte saisi qui indique la fin de la mention.
  - Score : niveau de confiance d'Amazon Comprehend quant à la précision du type d'entité.
  - GroupScore: le niveau de confiance d'Amazon Comprehend quant au fait que la mention est correctement regroupée avec d'autres mentions de la même entité.
  - Texte : texte de l'entité.
  - Type : type de l'entité. Pour tous les types d'entités pris en charge, consultezTypes d'entités.

Événements

Amazon Comprehend renvoie la liste des événements (des types d'événements pris en charge) détectés dans le texte saisi. Chaque événement revient avec les informations connexes suivantes :

- Type : type d'événement. Pour tous les types d'événements pris en charge, consultezTypes d’événements.
- Arguments : liste d'arguments liés à l'événement détecté. Un argument est constitué d'une entité associée à l'événement détecté. Le rôle de l'argument décrit la relation, par exemple qui a fait quoi, où et quand.
  - EntityIndex: valeur d'index identifiant une entité dans la liste des entités renvoyées par Amazon Comprehend pour cette analyse.
  - Rôle : type d'argument, qui décrit comment l'entité associée à cet argument est liée à l'événement. Pour tous les types d'arguments pris en charge, consultezTypes d'arguments.
  - Score : niveau de confiance d'Amazon Comprehend quant à la précision de la détection des rôles.
- Déclencheurs : liste des déclencheurs pour l'événement détecté. Un déclencheur est un mot ou une phrase unique qui indique la survenance de l'événement.

- BeginOffset: décalage de caractères dans le texte saisi qui indique le point de départ du déclencheur (le premier caractère est à la position 0).
- EndOffset: décalage de caractères dans le texte saisi qui indique où se termine le déclencheur.
- Score : niveau de confiance d'Amazon Comprehend quant à la précision de la détection.
- Texte : texte du déclencheur.
- GroupScore: le niveau de confiance d'Amazon Comprehend quant au fait que le déclencheur est correctement regroupé avec d'autres déclencheurs pour le même événement.
- Type : type d'événement indiqué par ce déclencheur.

Format des résultats de détection des événements

Lorsque votre tâche de détection d'événements est terminée, Amazon Comprehend écrit les résultats de l'analyse sur l'emplacement de sortie Amazon S3 que vous avez spécifié lorsque vous avez commencé la tâche.

Pour chaque événement détecté, la sortie fournit des détails au format suivant :

```
{
   "Entities": [
     {
       "Mentions": [
         {
           "BeginOffset": number,
           "EndOffset": number,
           "Score": number,
           "GroupScore": number,
           "Text": "string",
           "Type": "string"
         }, ...
       ]
     }, ...
   ],
   "Events": [
     {
       "Type": "string",
       "Arguments": [
         {
           "EntityIndex": number,
           "Role": "string",
           "Score": number
```

```
        }, ...
      ],
      "Triggers": [
        {
          "BeginOffset": number,
          "EndOffset": number,
          "Score": number,
          "Text": "string",
          "GroupScore": number,
          "Type": "string"
        }, ...
      ]
    }, ...
  ]
}
```

Types pris en charge pour les entités, les événements et les arguments

Types d'entités

Type	Description
DATE	Toute référence à une date ou à une heure, qu'elle soit spécifique ou générale.
INSTALLATION	Bâtiments, aéroports, autoroutes, ponts et autres structures humaines permanentes et améliorations immobilières.
LOCATION	Emplacements physiques tels que les rues, les villes, les États, les pays, les plans d'eau ou les coordonnées géographiques.
VALEUR_MONÉTAIRE	La valeur d'un objet en dollars américains ou dans une autre devise. La valeur peut être spécifique ou approximative.
ORGANISATION	Entreprises et autres groupes de personnes définis par une structure organisationnelle établie.

Type	Description
LA PERSONNE	Les noms ou surnoms d'individus ou de personnages fictifs.
TITRE DE LA PERSONNE	Tout titre décrivant une personne, qui est généralement une catégorie d'emploi (comme PDG) ou honorifique (comme M.).
QUANTITÉ	Un nombre ou une valeur et l'unité de mesure.
STOCK_CODE	Un symbole boursier, tel que AMZN, un numéro international d'identification des titres (ISIN), le Committee on Uniform Securities Identification Procedures (CUSIP) ou la Stock Exchange Daily Official List (SEDOL).

Types d’événements

Type	Description
FAILLITE	Procédure judiciaire impliquant une personne ou une entreprise incapable de rembourser ses dettes impayées.
EMPLOI	Se produit lorsqu'un employé est embauché, licencié, mis à la retraite ou change de statut d'emploi.
ACQUISITION D'ENTREPRISE	Se produit lorsqu'une entreprise obtient la possession de la plupart ou de la totalité des actions ou des actifs physiques d'une autre société pour prendre le contrôle de cette société.
INVESTISSEMENT_GÉNÉRAL	Se produit lorsqu'une personne ou une entreprise achète un actif dans la perspective

Type	Description
	de générer des revenus ou une appréciation futurs.
FUSION_ENTREPRISE	Se produit lorsque deux entreprises ou plus s'unissent pour créer une nouvelle entité juridique.
INTRODUCTION EN BOURSE	Une offre publique initiale (IPO) d'actions d'une société privée au public dans le cadre d'une nouvelle émission d'actions.
PROBLÈME_DROIT	Un ensemble de droits offerts aux actionnaires existants pour acheter des actions supplémentaires, appelés bons de souscription, proportionnellement à leurs avoirs existants.
OFFRE_SECONDAIRE	Offre de titres par un actionnaire d'une société.
SHELF_OFFERING	Une disposition de la Securities and Exchange Commission (SEC) qui permet à un émetteur d'enregistrer une nouvelle émission de titre et d'en vendre des parties sur une certaine période sans réenregistrer le titre ni encourir de pénalités. Également connu sous le nom d'enregistrement en rayon.
OFFRE_D'OFFRES	Offre d'achat d'une partie ou de la totalité des actions détenues par les actionnaires d'une entreprise.
STOCK_SPLIT	Se produit lorsque le conseil d'administration d'une entreprise augmente le nombre d'actions en circulation en émettant davantage d'actions aux actionnaires actuels. Cet événement s'applique également aux scissions inversées d'actions.

Types d'arguments

Types d'arguments en cas de FAILLITE

Type d’argument	Description
FILTRE	La personne ou l'entreprise qui déclare faillite.
DATE	La date ou l'heure de la faillite.
ENDROIT	Lieu ou établissement où (ou le plus proche de l'endroit) la faillite a eu lieu.

Types d'arguments en faveur de l'EMPLOI

Type	Description
EMPLOYÉ	La personne employée par une entreprise.
TITLE_EMPLOYÉ	Titre de l'employé.
EMPLOYEUR	La personne ou l'entreprise qui emploie l'employé.
START_DATE	La date ou l'heure de début de l'emploi.
DATE DE FIN	La date ou l'heure de fin de l'emploi.

Types d'arguments pour CORPORATE_ACQUISTION, INVESTMENT_GENERAL

Type	Description
MONTANT	La valeur monétaire associée à la transaction.
ENTITÉ INVESTIE	La personne ou l'entreprise associée à l'investissement.
INVESTISSEUR	La personne ou l'entreprise qui investit dans l'actif.

Type	Description
DATE	Date ou heure de l'acquisition ou de l'investissement.
ENDROIT	Lieu où (ou le plus proche de l'endroit) l'acquisition ou l'investissement a eu lieu.

Types d'arguments pour CORPORATE_MERGER

Type	Description
DATE	Date ou heure de la fusion.
NOUVELLE_ENTREPRISE	La nouvelle entité juridique issue de la fusion.
PARTICIPANT	La société impliquée dans la fusion.

Types d'arguments pour IPO, RIGHTS_ISSUE, SECONDARY_OFFERING, SHELF_OFFERING, TENDER_OFFERING

Type	Description
DATE_D'EXPIRATION	Date ou heure d'expiration de l'offre.
INVESTISSEUR	La personne ou l'entreprise qui investit dans l'actif.
DESTINATAIRE	La personne ou l'entreprise qui reçoit l'offre.
MONTANT_DE L'OFFRE	La valeur monétaire associée à l'offre.
DATE_D'OFFRE	La date ou l'heure de l'offre.
QUI OFFRE	La personne ou l'entreprise à l'origine de l'offre.
VALEUR_TOTAL_DE L'OFFREUR	La valeur monétaire totale associée à l'offre.
DATE_D'ENREGISTREMENT	Date ou heure d'enregistrement de l'offre.

Type	Description
AGENT DE VENTE	La personne ou l'entreprise qui facilite la vente de l'offre.
PRIX_ACTION	La valeur monétaire associée au cours de l'action.
PARTAGER_QUANTITÉ	Le nombre d'actions associées à l'offre.
SOUSCRIPTEURS	La société associée à la souscription de l'offre.

Types d'arguments pour STOCK_SPLIT

Type	Description
ENTREPRISE	La société émettrice des actions s'est scindée.
DATE	Date ou heure du fractionnement des actions.
RATIO DE DIVISION	Le ratio entre l'augmentation du nombre de nouvelles actions en circulation et le nombre actuel d'actions avant le fractionnement des actions.

Phrases clés

Une phrase clé est une chaîne contenant un syntagme nominal qui décrit une chose en particulier. Il se compose généralement d'un nom et des modificateurs qui le distinguent. Par exemple, « jour » est un nom ; « une belle journée » est un nom qui inclut un article (« a ») et un adjectif (« beau »). Chaque phrase clé inclut un score qui indique le niveau de confiance d'Amazon Comprehend quant au fait que la chaîne est une phrase nominale. Vous pouvez utiliser le score pour déterminer si la détection est suffisamment fiable pour votre application.

Les opérations de détection des phrases clés peuvent être effectuées dans l'un des principaux langages pris en charge par Amazon Comprehend. Tous les documents doivent être rédigés dans la même langue.

Vous pouvez utiliser l'une des opérations suivantes pour détecter des phrases clés dans un document ou un ensemble de documents.

- DetectKeyPhrases
- BatchDetectKeyPhrases
- StartKeyPhrasesDetectionJob

Les opérations renvoient une liste d'KeyPhraseobjets, un pour chaque phrase clé du document. L'`BatchDetectKeyPhrases`opération renvoie une liste d'`KeyPhrase`objets, un pour chaque document du lot. L'`StartKeyPhrasesDetectionJob`opération démarre une tâche asynchrone qui produit un fichier contenant une liste d'`KeyPhrase`objets pour chaque document de la tâche.

L'exemple suivant est la réponse de l'`DetectKeyPhrases`opération.

```
{
    "LanguageCode": "en",
    "KeyPhrases": [
        {
            "Text": "today",
            "Score": 0.89,
            "BeginOffset": 14,
            "EndOffset": 19
        },
        {
            "Text": "Seattle",
            "Score": 0.91,
            "BeginOffset": 23,
            "EndOffset": 30
        }
    ]
}
```

Langue dominante

Vous pouvez utiliser Amazon Comprehend pour examiner le texte afin de déterminer la langue dominante. Amazon Comprehend identifie la langue à l'aide des identifiants de la RFC 5646. S'il existe un identifiant ISO 639-1 à 2 lettres, avec une sous-étiquette régionale si nécessaire, il l'utilise. Sinon, il utilise le code à 3 lettres ISO 639-2.

Pour plus d'informations sur la RFC 5646, consultez la section Tags pour identifier les langues sur le site Web de l'IETF Tools.

La réponse inclut un score qui indique le niveau de confiance d'Amazon Comprehend quant au fait qu'une langue particulière est la langue dominante du document. Chaque score est indépendant des autres scores. Le score n'indique pas qu'une langue constitue un pourcentage particulier d'un document.

Si un document long (tel qu'un livre) contient plusieurs langues, vous pouvez le diviser en petits morceaux et exécuter l'`DetectDominantLanguage`opération sur chaque élément. Vous pouvez ensuite agréger les résultats pour déterminer le pourcentage de chaque langue dans le document le plus long.

La détection du langage par Amazon Comprehend présente les limites suivantes :

- Il ne prend pas en charge la détection de la langue phonétique. Par exemple, il ne détecte pas « arigato » en japonais ou « nihao » en chinois.
- Il peut être difficile de distinguer des paires linguistiques proches, telles que l'indonésien et le malais, ou le bosniaque, le croate et le serbe.
- Pour de meilleurs résultats, saisissez au moins 20 caractères de texte.

Amazon Comprehend détecte les langues suivantes.

Code	Langue
af	Afrikaans
am	Amharique
ar	Arabe
as	Assamais
az	Azerbaïdjanais
ba	Bachkir
be	Biélorusse

Code	Langue
bn	Bengali
bs	Bosniaque
bg	Bulgare
ca	Catalan
ceb	Cebuano
cs	Tchèque
cv	Tchouvache
cy	Gallois
da	Danois
de	Allemand
el	Grec
en	Anglais
eo	espéranto
et	Estonian
eu	Basque
fa	Persan
fi	Finnois
fr	Français
gd	Gaélique écossais
ga	irlandais

Code	Langue
gl	Galicien
gu	Gujarati
ht	Haïtien
he	Hébreu
ha	Haoussa
hi	Hindi
hr	Croate
hu	Hongrois
hy	Arménien
ilo	Iloko
id	Indonésien
is	Islandais
it	Italien
jv	Javanais
ja	Japonais
kn	Kannada
ka	Géorgien
kk	Kazakh
km	Khmer central
ky	kirghize

Code	Langue
ko	Coréen
ku	Kurde
lo	Laos
la	Latin
lv	Letton
lt	Lituanien
lb	Luxembourgeois
ml	Malayalam
mt	Maltais
mr	Marathi
mk	Macédonien
mg	Malgache
mn	Mongol
ms	Malais
my	Birman
ne	Népalais
new	Newari
nl	Néerlandais
no	Norvégien
or	Oriya

Code	Langue
om	Oromo
pa	Pendjabi
pl	Polonais
pt	Portugais
ps	Pushto
qu	Quechua
ro	Roumain
ru	Russe
sa	sanskrit
si	Singhalais
sk	Slovaque
sl	Slovène
sd	Sindhi
so	Somali
es	Espagnol
sq	Albanais
sr	Serbe
su	Soundanais
sw	Swahili
sv	Suédois

Code	Langue
ta	Tamoul
tt	Tatar
te	Télougou
tg	Tadjik
tl	Tagalog
th	Thaï
tk	Turkmène
tr	Turc
ug	Ouïghour
uk	Ukrainien
ur	Urdu
uz	Ouzbek
vi	Vietnamien
yi	yiddish
yo	Yoruba
zh	Chinois (simplifié)
zh-TW	Chinois (Traditionnel)

Vous pouvez utiliser l'une des opérations suivantes pour détecter la langue dominante dans un document ou un ensemble de documents.

- DetectDominantLanguage
- BatchDetectDominantLanguage

- StartDominantLanguageDetectionJob

L'`DetectDominantLanguage`opération renvoie un DominantLanguageobjet. L'`BatchDetectDominantLanguage`opération renvoie une liste d'`DominantLanguage`objets, un pour chaque document du lot. L'`StartDominantLanguageDetectionJob`opération démarre une tâche asynchrone qui produit un fichier contenant une liste d'`DominantLanguage`objets, un pour chaque document de la tâche.

L'exemple suivant est la réponse de l'`DetectDominantLanguage`opération.

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9793661236763
        }
    ]
}
```

Sentiment

Utilisez Amazon Comprehend pour déterminer le contenu des documents texte codés en UTF-8. Par exemple, vous pouvez utiliser l'analyse des sentiments pour déterminer le sentiment des commentaires sur un article de blog afin de déterminer si vos lecteurs ont aimé le billet.

Vous pouvez déterminer le sentiment à l'égard des documents rédigés dans l'une des principales langues prises en charge par Amazon Comprehend. Tous les documents d'une même tâche doivent être rédigés dans la même langue.

La détermination du sentiment renvoie les valeurs suivantes :

- Positif — Le texte exprime un sentiment globalement positif.
- Négatif — Le texte exprime un sentiment général négatif.
- Mixte — Le texte exprime à la fois des sentiments positifs et négatifs.
- Neutre — Le texte n'exprime aucun sentiment positif ou négatif.

Vous pouvez utiliser l'une des opérations d'API suivantes pour détecter le sentiment d'un document ou d'un ensemble de documents.

- DetectSentiment
- BatchDetectSentiment
- StartSentimentDetectionJob

Les opérations renvoient le sentiment le plus probable pour le texte et les scores pour chacun des sentiments. Le score représente la probabilité que le sentiment ait été correctement détecté. Par exemple, dans l'exemple ci-dessous, il est probable que le texte exprime un `Positive` sentiment à 95 %. Il y a moins de 1 % de probabilité que le texte exprime un `Negative` sentiment. Vous pouvez utiliser le `SentimentScore` pour déterminer si la précision de la détection répond aux besoins de votre application.

L'`DetectSentiment`opération renvoie un objet contenant le sentiment détecté et un SentimentScoreobjet. L'`BatchDetectSentiment`opération renvoie une liste de sentiments et `SentimentScore` d'objets, une pour chaque document du lot. L'`StartSentimentDetectionJob`opération lance une tâche asynchrone qui produit un fichier contenant une liste de sentiments et d'`SentimentScore`objets, une pour chaque document de la tâche.

L'exemple suivant est la réponse de l'`DetectSentiment`opération.

```
{
"SentimentScore": {
        "Mixed": 0.030585512690246105,
        "Positive": 0.94992071056365967,
        "Neutral": 0.0141543131828308,
        "Negative": 0.00893945890665054
    },
    "Sentiment": "POSITIVE",
    "LanguageCode": "en"
}
```

Sentiment ciblé

Le sentiment ciblé fournit une compréhension précise des sentiments associés à des entités spécifiques (telles que des marques ou des produits) dans vos documents d'entrée.

La différence entre le sentiment ciblé et le sentiment réside dans le niveau de granularité des données de sortie. L'analyse des sentiments détermine le sentiment dominant pour chaque document

d'entrée, mais ne fournit pas de données pour une analyse plus approfondie. L'analyse ciblée des sentiments détermine le sentiment au niveau de l'entité pour des entités spécifiques dans chaque document d'entrée. Vous pouvez analyser les données de sortie pour déterminer les produits et services spécifiques qui reçoivent des commentaires positifs ou négatifs.

Par exemple, dans une série de critiques de restaurants, un client fournit le commentaire suivant : « Les tacos étaient délicieux et le personnel était sympathique. » L'analyse de cette revue produit les résultats suivants :

- L'analyse des sentiments détermine si le sentiment général de chaque critique de restaurant est positif, négatif, neutre ou mitigé. Dans cet exemple, le sentiment général est positif.
- L'analyse ciblée des sentiments détermine le sentiment à l'égard des entités et des attributs du restaurant mentionnés par les clients dans les avis. Dans cet exemple, le client a fait des commentaires positifs à propos des « tacos » et du « personnel ».

Le sentiment ciblé fournit les résultats suivants pour chaque tâche d'analyse :

- Identité des entités mentionnées dans les documents.
- Classification du type d'entité pour chaque mention d'entité.
- Le sentiment et un score de sentiment pour chaque entité mentionnée.
- Groupes de mentions (groupes de coréférence) correspondant à une seule entité.

Vous pouvez utiliser la console ou l'API pour effectuer une analyse des sentiments ciblée. La console et l'API prennent en charge l'analyse en temps réel et l'analyse asynchrone pour un sentiment ciblé.

Amazon Comprehend prend en charge le sentiment ciblé pour les documents rédigés en anglais.

Pour plus d'informations sur le sentiment ciblé, y compris un didacticiel, voir Extraire un sentiment granulaire dans un texte avec Amazon Comprehend Targeted Sentiment sur AWS le blog de machine learning.

Rubriques

- Types d'entités
- Groupe de coréférence
- Organisation des fichiers de sortie
- Analyse en temps réel à l'aide de la console

- Exemple de production de sentiments ciblés

Types d'entités

Le sentiment ciblé identifie les types d'entités suivants. Il attribue le type d'entité OTHER si l'entité n'appartient à aucune autre catégorie. Chaque entité mentionnée dans le fichier de sortie inclut le type d'entité, tel que`"Type": "PERSON"`.

Définitions des types d'entités

Type d'entité	Définition
LA PERSONNE	Les exemples incluent des individus, des groupes de personnes, des surnoms, des personnages fictifs et des noms d'animaux.
LOCATION	Emplacements géographiques tels que les pays, les villes, les états, les adresses, les formations géologiques, les plans d'eau, les sites naturels et les sites astronomiques.
ORGANISATION	Les exemples incluent les gouvernements, les entreprises, les équipes sportives et les religions.
INSTALLATION	Bâtiments, aéroports, autoroutes, ponts et autres structures humaines permanentes et améliorations immobilières.
MARQUE	Organisation, groupe ou producteur d'un article commercial ou d'une gamme de produits spécifique.
ARTICLE_COMMERCIAL	Tout article non générique achetable ou acquérable, y compris les véhicules, et les gros produits fabriqués par un seul article.
FILM	Un film ou une émission de télévision. L'entité peut être le nom complet, un surnom ou un sous-titre.
MUSIQUE	Une chanson, complète ou partielle. Également, des collections de créations musicales individuelles, telles qu'un album ou une anthologie.
RÉSERVER	Un livre, publié par un professionnel ou publié à compte d'auteur.
LOGICIEL	Un produit logiciel officiellement publié.

Type d'entité	Définition
JEU	Un jeu, tel que des jeux vidéo, des jeux de société, des jeux courants ou des sports.
TITRE_PERSONNEL	Titres officiels et distinctions honorifiques tels que président, doctorat ou Dr.
ÉVÉNEMENT	Les exemples incluent un festival, un concert, une élection, une guerre, une conférence et un événement promotionnel.
DATE	Toute référence à une date ou à une heure, qu'elle soit spécifique ou générale, absolue ou relative.
QUANTITÉ	Toutes les mesures ainsi que leurs unités (devise, pourcentage, nombre, octets, etc.).
ATTRIBUTE	Attribut, caractéristique ou trait d'une entité, tel que la « qualité » d'un produit, le « prix » d'un téléphone ou la « vitesse » d'un processeur.
OTHER	Entités qui n'appartiennent à aucune des autres catégories.

Groupe de coréférence

Le sentiment ciblé identifie les groupes de coréférence dans chaque document d'entrée. Un groupe de coréférence est un groupe de mentions dans un document qui correspondent à une entité du monde réel.

Example

Dans l'exemple suivant d'avis client, « spa » est l'entité qui possède un type d'entité`FACILITY`. L'entité possède deux mentions supplémentaires sous forme de pronom (« it »).

Organisation des fichiers de sortie

La tâche d'analyse des sentiments ciblée crée un fichier de sortie texte JSON. Le fichier contient un objet JSON pour chacun des documents d'entrée. Chaque objet JSON contient les champs suivants :

- Entités : tableau d'entités présentes dans le document.
- Fichier : nom de fichier du document d'entrée.
- Ligne — Si le fichier d'entrée contient un document par ligne, Entities contient le numéro de ligne du document dans le fichier.

> **Note**
>
> Si le sentiment ciblé n'identifie aucune entité dans le texte saisi, il renvoie un tableau vide comme résultat Entities.

L'exemple suivant montre les entités d'un fichier d'entrée comportant trois lignes de saisie. Le format d'entrée est ONE_DOC_PER_LINE, de sorte que chaque ligne d'entrée est un document.

```
{ "Entities":[
    {entityA},
    {entityB},
    {entityC}
    ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 0
}
```

```
{ "Entities": [
    {entityD},
    {entityE}
  ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 1
}
{ "Entities": [
    {entityF},
    {entityG}
    ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 2
}
```

Une entité du tableau Entities inclut un regroupement logique (appelé groupe de coréférence) des mentions d'entités détectées dans le document. La structure globale de chaque entité est la suivante :

```
{"DescriptiveMentionIndex": [0],
  "Mentions": [
     {mentionD},
     {mentionE}
    ]
}
```

Une entité contient les champs suivants :

- Mentions : tableau de mentions de l'entité dans le document. Le tableau représente un groupe de coréférence. Veuillez consulter the section called “Groupe de coréférence” pour obtenir un exemple. L'ordre des mentions dans le tableau Mentions est l'ordre de leur emplacement (décalage) dans le document. Chaque mention inclut le score de sentiment et le score de groupe pour cette mention. Le score du groupe indique le niveau de confiance selon lequel ces mentions appartiennent à la même entité.
- DescriptiveMentionIndex— Un ou plusieurs index du tableau Mentions fournissant le meilleur nom pour le groupe d'entités. Par exemple, une entité peut avoir trois mentions avec les valeurs de texte « ABC Hotel », « ABC Hotel » et « it ». Le meilleur nom est « ABC Hotel », qui a une DescriptiveMentionIndex valeur de [0,1].

Chaque mention inclut les champs suivants

- BeginOffset— Le décalage dans le texte du document à l'endroit où la mention commence.
- EndOffset— Le décalage dans le texte du document à la fin de la mention.
- GroupScore— La certitude que toutes les entités mentionnées dans le groupe sont liées à la même entité.
- Texte : texte du document qui identifie l'entité.
- Type : type de l'entité. Amazon Comprehend prend en charge différents types d'entités.
- Score : modèle de confiance quant à la pertinence de l'entité. La plage de valeurs est comprise entre zéro et un, 1 correspondant au niveau de confiance le plus élevé.
- MentionSentiment— Contient le sentiment et le score de sentiment associés à la mention.
- Sentiment — Le sentiment qui sous-tend la mention. Les valeurs incluent : POSITIF, NEUTRE, NÉGATIF et MIXTE.
- SentimentScore— Fournit un modèle de confiance pour chacun des sentiments possibles. La plage de valeurs est comprise entre zéro et un, 1 correspondant au niveau de confiance le plus élevé.

Les valeurs Sentiment ont la signification suivante :

- Positif — La mention de l'entité exprime un sentiment positif.
- Négatif — La mention de l'entité exprime un sentiment négatif.
- Mixte — La mention de l'entité exprime à la fois des sentiments positifs et négatifs.
- Neutre — L'entité mentionnée n'exprime aucun sentiment positif ou négatif.

Dans l'exemple suivant, une entité n'a qu'une seule mention dans le document d'entrée, DescriptiveMentionIndex donc zéro (la première mention dans le tableau Mentions). L'entité identifiée est une PERSONNE portant le nom « I ». Le score de sentiment est neutre.

```
{"Entities":[
  {
    "DescriptiveMentionIndex": [0],
    "Mentions": [
      {
       "BeginOffset": 0,
       "EndOffset": 1,
       "Score": 0.999997,
       "GroupScore": 1,
```

```
      "Text": "I",
      "Type": "PERSON",
      "MentionSentiment": {
        "Sentiment": "NEUTRAL",
        "SentimentScore": {
          "Mixed": 0,
          "Negative": 0,
          "Neutral": 1,
          "Positive": 0
        }
      }
    }
   ]
  }
 ],
 "File": "Input.txt",
 "Line": 0
}
```

Analyse en temps réel à l'aide de la console

Vous pouvez utiliser la console Amazon Comprehend pour l'exécuter the section called “Sentiment ciblé” en temps réel. Utilisez le texte d'exemple ou collez votre propre texte dans la zone de saisie, puis choisissez Analyser.

Dans le panneau Insights, la console affiche trois vues de l'analyse des sentiments ciblée :

- Texte analysé : affiche le texte analysé et souligne chaque entité. La couleur du soulignement indique la valeur de sentiment (positive, neutre, négative ou mixte) attribuée par l'analyse à l'entité. La console affiche les mappages de couleurs dans le coin supérieur droit de la zone de texte analysée. Si vous passez le curseur sur une entité, la console affiche un panneau contextuel contenant les valeurs d'analyse (type d'entité, score de sentiment) pour l'entité.
- Résultats — Affiche un tableau contenant une ligne pour chaque mention d'entité identifiée dans le texte. Pour chaque entité, le tableau indique l'entité et le score de l'entité. La ligne inclut également le sentiment principal et le score pour chaque valeur de sentiment. S'il existe plusieurs mentions de la même entité, connue sous le nom de athe section called “Groupe de coréférence”, le tableau affiche ces mentions sous la forme d'un ensemble démontable de lignes associées à l'entité principale.

  Si vous survolez une ligne d'entité dans le tableau des résultats, la console met en évidence la mention de l'entité dans le panneau de texte analysé.

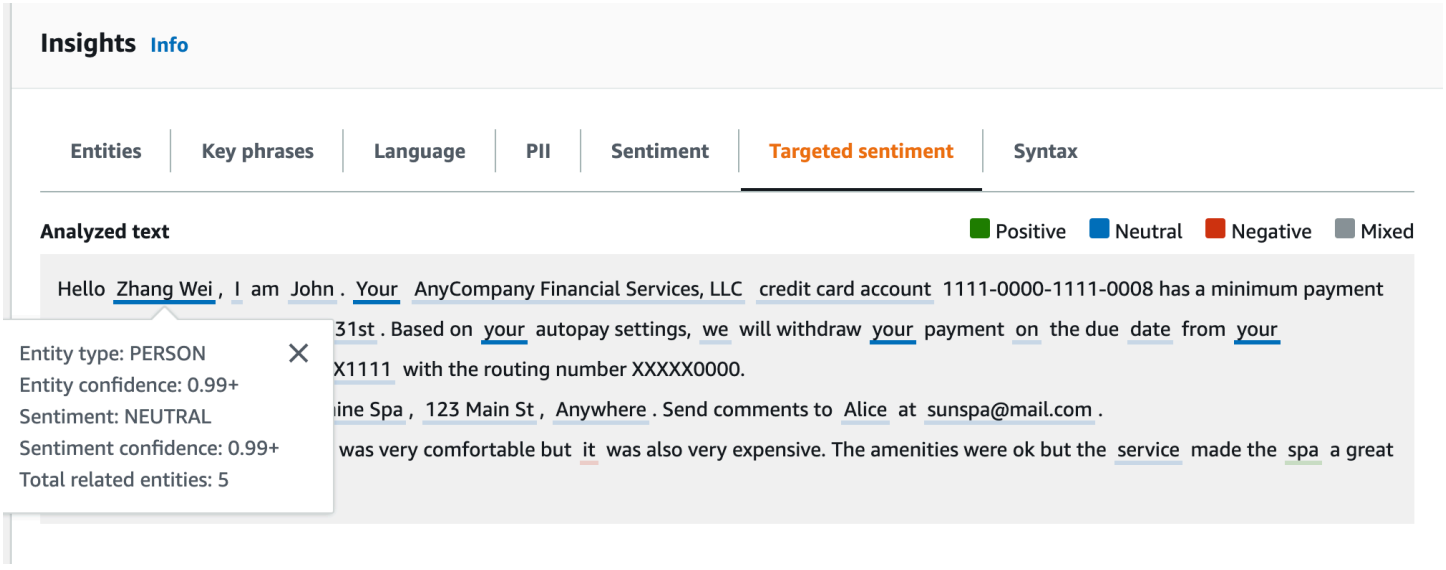
- Intégration des applications : affiche les valeurs des paramètres de la demande d'API et la structure de l'objet JSON renvoyé dans la réponse de l'API. Pour une description des champs de l'objet JSON, consultezthe section called “Organisation des fichiers de sortie”.

Exemple d'analyse en temps réel sur console

Cet exemple utilise le texte suivant comme entrée, qui est le texte d'entrée par défaut fourni par la console.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account
 1111-0000-1111-0008 has a minimum payment
  of $24.53 that is due by July 31st. Based on your autopay settings, we will withdraw
 your payment on the due date from your
  bank account number XXXXXX1111 with the routing number XXXXX0000.
  Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at
 sunspa@mail.com.
  I enjoyed visiting the spa. It was very comfortable but it was also very expensive.
 The amenities were ok but the service made
  the spa a great experience.
```

Le panneau de texte analysé affiche le résultat suivant pour cet exemple. Passez votre souris sur le texte `Zhang Wei` pour afficher le panneau contextuel de cette entité.



Le tableau des résultats fournit des détails supplémentaires sur chaque entité, notamment le score de l'entité, le sentiment principal et le score de chaque sentiment.

▼ Results

Search 1 2

Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne
Zhang Wei (5)	PERSON	-	NEUTRAL	-	-
John (3)	PERSON	-	NEUTRAL	-	-
AnyCompany Financial Services, LLC (2)	ORGANIZATION	-	NEUTRAL	-	-
credit card account	OTHER	0.99+	NEUTRAL	0.00	0.0
$24.53	QUANTITY	0.99+	NEUTRAL	0.00	0.0
by July 31st (3)	DATE	-	NEUTRAL	-	-
bank account	OTHER	0.99+	NEUTRAL	0.00	0.0
XXXXXX1111	OTHER	0.51	NEUTRAL	0.00	0.0
Customer	PERSON	0.98	NEUTRAL	0	0
Sunshine Spa (5)	FACILITY	-	MIXED	-	-

Dans notre exemple, l'analyse ciblée des sentiments reconnaît que chaque mention de vous dans le texte saisi est une référence à l'entité personnelle Zhang Wei. La console affiche ces mentions sous la forme d'un ensemble de lignes démontables associées à l'entité principale.

▼ Results

Search 1 2

Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne
Zhang Wei (5)	PERSON	-	NEUTRAL	-	-
your	PERSON	0.99+	NEUTRAL	0	0
your	PERSON	0.67	NEUTRAL	0	0
Your	ORGANIZATION	0.94	NEUTRAL	0	0
your	PERSON	0.99+	NEUTRAL	0	0
Zhang Wei	PERSON	0.99+	NEUTRAL	0.00	0

Le panneau d'intégration des applications affiche l'objet JSON généré par l' DetectTargetedSentiment API. Consultez la section suivante pour un exemple complet.

Exemple de production de sentiments ciblés

L'exemple suivant montre le fichier de sortie d'une tâche d'analyse des sentiments ciblée. Le fichier d'entrée se compose de trois documents simples :

```
The burger was very flavorful and the burger bun was excellent. However, customer
 service was slow.
My burger was good, and it was warm. The burger had plenty of toppings.
The burger was cooked perfectly but it was cold. The service was OK.
```

L'analyse ciblée des sentiments de ce fichier d'entrée produit le résultat suivant.

```
 {"Entities":[
   {
     "DescriptiveMentionIndex": [
       0
     ],
     "Mentions": [
       {
         "BeginOffset": 4,
         "EndOffset": 10,
         "Score": 0.999991,
         "GroupScore": 1,
         "Text": "burger",
         "Type": "OTHER",
         "MentionSentiment": {
           "Sentiment": "POSITIVE",
           "SentimentScore": {
             "Mixed": 0,
             "Negative": 0,
             "Neutral": 0,
             "Positive": 1
           }
         }
       }
     ]
   },
   {
     "DescriptiveMentionIndex": [
       0
     ],
     "Mentions": [
       {
```

```
          "BeginOffset": 38,
          "EndOffset": 44,
          "Score": 1,
          "GroupScore": 1,
          "Text": "burger",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0.000005,
              "Negative": 0.000005,
              "Neutral": 0.999591,
              "Positive": 0.000398
            }
          }
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 45,
          "EndOffset": 48,
          "Score": 0.961575,
          "GroupScore": 1,
          "Text": "bun",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
              "Mixed": 0.000327,
              "Negative": 0.000286,
              "Neutral": 0.050269,
              "Positive": 0.949118
            }
          }
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
```

```
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 73,
          "EndOffset": 89,
          "Score": 0.999988,
          "GroupScore": 1,
          "Text": "customer service",
          "Type": "ATTRIBUTE",
          "MentionSentiment": {
            "Sentiment": "NEGATIVE",
            "SentimentScore": {
              "Mixed": 0.000001,
              "Negative": 0.999976,
              "Neutral": 0.000017,
              "Positive": 0.000006
            }
          }
        }
      ]
    }
  ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 0
}
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 0,
          "EndOffset": 2,
          "Score": 0.99995,
          "GroupScore": 1,
          "Text": "My",
          "Type": "PERSON",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
```

```
            "Negative": 0,
            "Neutral": 1,
            "Positive": 0
          }
        }
      }
    ]
  },
  {
    "DescriptiveMentionIndex": [
      0,
      2
    ],
    "Mentions": [
      {
        "BeginOffset": 3,
        "EndOffset": 9,
        "Score": 0.999999,
        "GroupScore": 1,
        "Text": "burger",
        "Type": "OTHER",
        "MentionSentiment": {
          "Sentiment": "POSITIVE",
          "SentimentScore": {
            "Mixed": 0.000002,
            "Negative": 0.000001,
            "Neutral": 0.000003,
            "Positive": 0.999994
          }
        }
      },
      {
        "BeginOffset": 24,
        "EndOffset": 26,
        "Score": 0.999756,
        "GroupScore": 0.999314,
        "Text": "it",
        "Type": "OTHER",
        "MentionSentiment": {
          "Sentiment": "POSITIVE",
          "SentimentScore": {
            "Mixed": 0,
            "Negative": 0.000003,
            "Neutral": 0.000006,
```

```
            "Positive": 0.999991
          }
        }
      },
      {
        "BeginOffset": 41,
        "EndOffset": 47,
        "Score": 1,
        "GroupScore": 0.531342,
        "Text": "burger",
        "Type": "OTHER",
        "MentionSentiment": {
          "Sentiment": "POSITIVE",
          "SentimentScore": {
            "Mixed": 0.000215,
            "Negative": 0.000094,
            "Neutral": 0.00008,
            "Positive": 0.999611
          }
        }
      }
    ]
  },
  {
    "DescriptiveMentionIndex": [
      0
    ],
    "Mentions": [
      {
        "BeginOffset": 52,
        "EndOffset": 58,
        "Score": 0.965462,
        "GroupScore": 1,
        "Text": "plenty",
        "Type": "QUANTITY",
        "MentionSentiment": {
          "Sentiment": "NEUTRAL",
          "SentimentScore": {
            "Mixed": 0,
            "Negative": 0,
            "Neutral": 1,
            "Positive": 0
          }
        }
```

```
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 62,
          "EndOffset": 70,
          "Score": 0.998353,
          "GroupScore": 1,
          "Text": "toppings",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0,
              "Neutral": 0.999964,
              "Positive": 0.000036
            }
          }
        }
      ]
    }
  ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 1
}
{
  "Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 4,
          "EndOffset": 10,
          "Score": 1,
          "GroupScore": 1,
          "Text": "burger",
```

```
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
              "Mixed": 0.001515,
              "Negative": 0.000822,
              "Neutral": 0.000243,
              "Positive": 0.99742
            }
          }
        },
        {
          "BeginOffset": 36,
          "EndOffset": 38,
          "Score": 0.999843,
          "GroupScore": 0.999661,
          "Text": "it",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "NEGATIVE",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0.999996,
              "Neutral": 0.000004,
              "Positive": 0
            }
          }
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 53,
          "EndOffset": 60,
          "Score": 1,
          "GroupScore": 1,
          "Text": "service",
          "Type": "ATTRIBUTE",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
```

```
            "SentimentScore": {
              "Mixed": 0.000033,
              "Negative": 0.000089,
              "Neutral": 0.993325,
              "Positive": 0.006553
            }
          }
        }
      ]
    }
  ],
  "File": "TargetSentimentInputDocs.txt",
  "Line": 2
}
  }
```

Analyse syntaxique

Utilisez l'analyse syntaxique pour analyser les mots du document et renvoyer la partie du discours, ou fonction syntaxique, pour chaque mot du document. Vous pouvez identifier les noms, les verbes, les adjectifs, etc. dans votre document. Utilisez ces informations pour mieux comprendre le contenu de vos documents et pour comprendre la relation entre les mots du document.

Par exemple, vous pouvez rechercher les noms dans un document, puis rechercher les verbes associés à ces noms. Dans une phrase comme « Ma grand-mère a déménagé son canapé », vous pouvez voir les noms « grand-mère » et « canapé », ainsi que le verbe « déménagé ». Vous pouvez utiliser ces informations pour créer des applications d'analyse de texte pour les combinaisons de mots qui vous intéressent.

Pour démarrer l'analyse, Amazon Comprehend analyse le texte source pour trouver les différents mots du texte. Une fois le texte analysé, chaque mot se voit attribuer la partie du discours qu'il prend dans le texte source.

Amazon Comprehend peut identifier les parties suivantes du discours.

Jeton	Partie du discours
ADJ	Adjectif Mots qui modifient généralem ent des noms.

Jeton	Partie du discours
ADP	Adposition Le titre d'une phrase prépositionnelle ou postpositionnelle.
ADV	Adverbe Mots qui modifient généralement les verbes. Ils peuvent également modifier les adjectifs et autres adverbes.
AUX	auxiliaire Mots fonctionnels qui accompagnent le verbe d'une phrase verbale.
CCONJ	Conjonction de coordination Une conjonction de coordination relie les mots, les phrases ou les clauses d'une phrase sans les subordonner les uns aux autres.
CONJ	Conjonction Une conjonction relie des mots, des phrases ou des clauses dans une phrase.
DET	Déterminant Articles et autres mots qui spécifient une phrase nominale particulière.

Jeton	Partie du discours
INTJ	Interjection Mots utilisés comme exclamation ou partie d'une exclamation.
NOM	Nom Mots qui désignent une personne, un lieu, une chose, un animal ou une idée.
NUM	Chiffre Mots, généralement des déterminants, des adjectifs ou des pronoms, qui expriment un nombre.
O	Autre Mots auxquels il n'est pas possible d'attribuer une partie de la catégorie vocale.
PART	Particle Utilisez des mots associés à un autre mot ou à une autre phrase pour donner du sens.
TABLIER	Pronom Mots qui remplacent des noms ou des phrases nominales.

Jeton	Partie du discours
PROPN	Nom propre Nom qui est le nom d'un individu, d'un lieu ou d'un objet spécifique.
POINT	Ponctuation Caractères non alphabétiques qui délimitent le texte.
SCONJ	Conjonction subordonnée Conjonction qui joint une clause dépendante à une phrase. Un exemple de conjonction subordonnée est « parce que ».
SYM	Symbol Entités semblables à des mots, telles que le signe dollar ($) ou des symboles mathématiques.
VERBE	Verb Des mots qui signalent des événements et des actions.

Pour plus d'informations sur les parties du discours, consultez les balises Universal POS sur le site Web d'Universal Dependencies.

Les opérations renvoient des jetons qui identifient le mot et la partie du discours que le mot représente dans le texte. Chaque jeton représente un mot dans le texte source. Il indique l'emplacement du mot dans la source, la partie du discours qu'il occupe dans le texte, la certitude

d'Amazon Comprehend quant à l'identification correcte de la partie du discours et le mot qui a été analysé à partir du texte source.

Voici la structure de la liste des jetons de syntaxe. Un jeton de syntaxe est généré pour chaque mot du document.

```
{
   "SyntaxTokens": [
      {
         "BeginOffset": number,
         "EndOffset": number,
         "PartOfSpeech": {
            "Score": number,
            "Tag": "string"
         },
         "Text": "string",
         "TokenId": number
      }
   ]
}
```

Chaque jeton fournit les informations suivantes :

- `BeginOffset`et `EndOffset` —Indique l'emplacement du mot dans le texte d'entrée.
- `PartOfSpeech`—Fournit deux informations, l'une `Tag` qui identifie la partie du discours et l'autre qui indique `Score` qu'Amazon Comprehend Syntax est certaine que la partie du discours a été correctement identifiée.
- `Text`—Fournit le mot qui a été identifié.
- `TokenId`—Fournit un identifiant pour le jeton. L'identifiant est la position du jeton dans la liste des jetons.

Amazon Comprehend personnalisé

Vous pouvez personnaliser Amazon Comprehend en fonction de vos besoins spécifiques sans avoir les compétences requises pour créer des solutions NLP basées sur le machine learning. À l'aide de l'apprentissage automatique, ou AutoML, Comprehend Custom crée des modèles de PNL personnalisés en votre nom, en utilisant les données d'entraînement que vous fournissez.

Traitement des documents d'entrée : Amazon Comprehend prend en charge le traitement des documents en une étape pour une classification personnalisée et une reconnaissance d'entité personnalisée. Par exemple, vous pouvez saisir un mélange de documents en texte brut et de documents semi-structurés (tels que des documents PDF, des documents Microsoft Word et des images) dans une tâche d'analyse personnalisée. Pour de plus amples informations, veuillez consulter Traitement des documents.

Classification personnalisée — Créez des modèles de classification personnalisés (classificateurs) pour organiser vos documents dans vos propres catégories. Pour chaque étiquette de classification, fournissez un ensemble de documents qui représentent le mieux cette étiquette et entraînez votre classificateur sur celle-ci. Une fois formé, un classificateur peut être utilisé sur un certain nombre d'ensembles de documents non étiquetés. Vous pouvez utiliser la console pour une expérience sans code ou installer le dernier AWS SDK. Pour de plus amples informations, veuillez consulter Classification personnalisée.

Reconnaissance d'entités personnalisée : créez des modèles de reconnaissance d'entités personnalisés (outils de reconnaissance) capables d'analyser le texte pour détecter des termes spécifiques et des phrases basées sur des noms. Vous pouvez former des personnes chargées de la reconnaissance à extraire des termes tels que des numéros de police ou des phrases qui impliquent une escalade de la relation client. Pour entraîner le modèle, vous fournissez une liste des entités et un ensemble de documents qui les contiennent. Une fois le modèle entraîné, vous pouvez soumettre des tâches d'analyse par rapport à celui-ci pour en extraire les entités personnalisées. Pour de plus amples informations, veuillez consulter Reconnaissance d'entités personnalisée.

Modélisation des rubriques

Vous pouvez utiliser Amazon Comprehend pour examiner le contenu d'un ensemble de documents afin de déterminer des thèmes communs. Par exemple, vous pouvez fournir à Amazon Comprehend une collection d'articles de presse, qui déterminera les sujets, tels que le sport, la politique ou le divertissement. Le texte des documents n'a pas besoin d'être annoté.

Amazon Comprehend utilise un modèle d'apprentissage basé sur l'allocation de dirichlets latents pour déterminer les sujets d'un ensemble de documents. Il examine chaque document afin de déterminer le contexte et le sens d'un mot. L'ensemble de mots qui appartiennent fréquemment au même contexte dans l'ensemble du document constitue un sujet.

Un mot est associé à un sujet dans un document en fonction de la prévalence de ce sujet dans le document et de l'affinité du sujet avec le mot. Le même mot peut être associé à différents sujets dans différents documents en fonction de la distribution des sujets dans un document particulier.

Par exemple, le mot « glucose » dans un article qui parle principalement de sport peut être attribué au sujet « sport », tandis que le même mot dans un article sur la « médecine » sera attribué au sujet « médecine ».

Chaque mot associé à un sujet reçoit une pondération qui indique dans quelle mesure le mot aide à définir le sujet. Le poids indique combien de fois le mot apparaît dans le sujet par rapport aux autres mots du sujet, sur l'ensemble du document.

Pour obtenir des résultats aussi précis que possible, vous devez fournir à Amazon Comprehend le plus grand corpus possible sur lequel travailler. Pour un résultat optimal :

- Vous devez utiliser au moins 1 000 documents dans chaque tâche de modélisation de sujets.
- Chaque document doit comporter au moins 3 phrases.
- Si un document est principalement composé de données numériques, vous devez le supprimer du corpus.

La modélisation thématique est un processus asynchrone. Vous soumettez votre liste de documents à Amazon Comprehend à partir d'un compartiment Amazon S3 à l'StartTopicsDetectionJobaide de cette opération. La réponse est envoyée à un compartiment Amazon S3. Vous pouvez configurer les compartiments d'entrée et de sortie. Obtenez la liste des rubriques de modélisation des tâches que vous avez soumises à l'aide de l'ListTopicsDetectionJobsopération et consultez les informations relatives à une tâche utilisant cette DescribeTopicsDetectionJobopération. Le contenu livré aux compartiments Amazon S3 peut contenir du contenu client. Pour plus d'informations sur la suppression de données sensibles, consultez How Do I Empty an S3 Bucket? (Comment puis-je vider un compartiment S3 ?) ou How Do I Delete an S3 Bucket? (Comment supprimer un compartiment S3 ?).

Les documents doivent utiliser le format UTF-8. Vous pouvez soumettre vos documents de deux manières. Le tableau suivant répertorie les options.

Format	Description
Un seul document par fichier	Chaque fichier contient un document d'entrée. C'est la meilleure solution pour les collections de documents volumineux.
Un document par ligne	L'entrée est un seul fichier. Chaque ligne du fichier est considérée comme un document.

Format	Description
	C'est la meilleure solution pour les documents courts, tels que les publications sur les réseaux sociaux. Chaque ligne doit se terminer par une ligne d'alimentation (LF,\n), un retour en chariot (CR, \ r), ou les deux (CRLF, \ r\n). Le séparateur de ligne Unicode (u+2028) ne peut pas être utilisé pour terminer une ligne.

Pour plus d'informations, consultez le type de données InputDataConfig.

Une fois qu'Amazon Comprehend a traité votre collection de documents, il renvoie une archive compressée contenant deux fichiers, et`topic-terms.csv`. `doc-topics.csv` Pour plus d'informations sur le fichier de sortie, consultez OutputDataConfig.

Le premier fichier de sortie est une liste des rubriques de la collection. `topic-terms.csv` Pour chaque sujet, la liste inclut, par défaut, les principaux termes par sujet en fonction de leur poids. Par exemple, si vous fournissez à Amazon Comprehend un ensemble d'articles de journaux, celui-ci peut renvoyer ce qui suit pour décrire les deux premiers sujets de la collection :

Rubrique	Durée	Weight
000	équipe	0,118533
000	game	0,106072
000	player	0,031625
000	saison	0,023633
000	jouer	0,021118
000	cour	0,024454
000	entraîneur	0,016012

Rubrique	Durée	Weight
000	jeux	0,016191
000	football	0,015049
000	quarterback	0,014239
001	tasse	0,205236
001	nourriture	0,040686
001	minutes	0,036062
001	ajouter	0,029697
001	cuillère à soupe	0,028789
001	huile	0,021254
001	poivre	0,022205
001	cuillère à café	0,020040
001	vin	0,016588
001	sucre	0,015101

Les poids représentent une distribution de probabilité sur les mots d'un sujet donné. Comme Amazon Comprehend renvoie uniquement les 10 premiers mots pour chaque sujet, la somme des pondérations ne sera pas égale à 1,0. Dans les rares cas où un sujet contient moins de 10 mots, la somme des poids sera de 1,0.

Les mots sont triés en fonction de leur pouvoir discriminant en examinant leur occurrence dans tous les sujets. Cela correspond généralement à leur poids, mais dans certains cas, comme les mots « play » et « yard » dans le tableau, cela se traduit par un ordre différent du poids.

Vous pouvez spécifier le nombre de sujets à renvoyer. Par exemple, si vous demandez à Amazon Comprehend de renvoyer 25 sujets, les 25 sujets les plus importants de la collection seront renvoyés. Amazon Comprehend peut détecter jusqu'à 100 sujets dans une collection. Choisissez le nombre de

sujets en fonction de vos connaissances du domaine. Il faudra peut-être faire quelques essais pour arriver au bon numéro.

Le second fichier répertorie `doc-topics.csv` les documents associés à un sujet et la proportion du document qui traite du sujet. Si vous l'avez spécifié, `ONE_DOC_PER_FILE` le document est identifié par le nom du fichier. Si vous l'avez spécifié, `ONE_DOC_PER_LINE` le document est identifié par le nom du fichier et le numéro de ligne indexé à 0 dans le fichier. Par exemple, Amazon Comprehend peut renvoyer les informations suivantes pour un ensemble de documents soumis avec un seul document par fichier :

Document	Rubrique	Proportion
sample-doc1	000	0,999330137
sample-doc2	000	0,998532187
sample-doc3	000	0,998384574
...		
Exemple Docn	000	3,57E-04

Amazon Comprehend utilise les informations du jeu de données de listes de lemmatisation de MBM, qui est disponible ici sous la licence Open database (L) v1.0. ODb

Modes de traitement des documents

Amazon Comprehend prend en charge trois modes de traitement des documents. Le mode que vous choisissez dépend du nombre de documents que vous devez traiter et de la rapidité avec laquelle vous devez consulter les résultats :

- Synchrone avec un seul document : vous appelez Amazon Comprehend avec un seul document et recevez une réponse synchrone, envoyée immédiatement à votre application (ou à la console).
- Synchrone avec plusieurs documents : vous appelez l'API Amazon Comprehend avec une collection de 25 documents au maximum et vous recevez une réponse synchrone.
- Lot asynchrone : pour une grande collection de documents, placez les documents dans un compartiment Amazon S3 et lancez une tâche asynchrone (à l'aide d'opérations de console ou

d'API) pour analyser les documents. Amazon Comprehend stocke les résultats de l'analyse dans le compartiment ou le dossier S3 que vous spécifiez dans la demande.

Rubriques

- Traitement d'un seul document
- Traitement synchrone de plusieurs documents
- Traitement par lots asynchrone

Traitement d'un seul document

Les opérations sur un seul document sont des opérations synchrones qui renvoient les résultats de l'analyse du document directement à votre application. Utilisez les opérations synchrones sur un seul document lorsque vous créez une application interactive qui fonctionne sur un seul document à la fois.

Pour plus d'informations sur les opérations d'API synchrones, consultez Analyse en temps réel à l'aide des modèles intégrés (pour la console) etAnalyse en temps réel à l'aide de l'API.

Traitement synchrone de plusieurs documents

Lorsque vous souhaitez traiter plusieurs documents, vous pouvez utiliser les opérations d'`Batch*`API pour envoyer plusieurs documents à la fois à Amazon Comprehend. Vous pouvez envoyer jusqu'à 25 documents par demande. Amazon Comprehend renvoie une liste de réponses, une pour chaque document de la demande. Les demandes effectuées avec ces opérations sont synchrones. Votre application lance l'opération puis attend la réponse du service.

L'utilisation `Batch*` des opérations est identique à l'appel du document unique APIs pour chacun des documents de la demande. Leur utilisation APIs peut améliorer les performances de vos applications.

L'entrée de chacun d'entre eux APIs est une structure JSON contenant les documents à traiter. Pour toutes les opérations`BatchDetectDominantLanguage`, à l'exception de la langue de saisie, vous devez définir la langue de saisie. Vous ne pouvez définir qu'une seule langue de saisie pour chaque demande. Par exemple, ce qui suit est l'entrée de l'`BatchDetectEntities`opération. Il contient deux documents et est en anglais.

```
{
   "LanguageCode": "en",
```

```
   "TextList": [
      "I have been living in Seattle for almost 4 years",
      "It is raining today in Seattle"
   ]
}
```

La réponse d'une `Batch*` opération contient deux listes, la `ResultList` et la`ErrorList`. `ResultList`Il contient un enregistrement pour chaque document traité avec succès. Le résultat pour chaque document de la demande est identique à celui que vous obtiendriez si vous exécutiez une seule opération sur le document. Un index est attribué aux résultats de chaque document en fonction de l'ordre des documents dans le fichier d'entrée. La réponse de l'`BatchDetectEntities`opération est la suivante :

```
{
   "ResultList"  : [
      {
         "Index": 0,
         "Entities": [
            {
               "Text": "Seattle",
               "Score": 0.95,
               "Type": "LOCATION",
               "BeginOffset": 22,
               "EndOffset": 29
            },
            {
               "Text": "almost 4 years",
               "Score": 0.89,
               "Type": "QUANTITY",
               "BeginOffset": 34,
               "EndOffset": 48
            }
         ]
      },
      {
         "Index": 1,
         "Entities": [
            {
              "Text": "today",
              "Score": 0.87,
              "Type": "DATE",
              "BeginOffset": 14,
```

```
              "EndOffset": 19
            },
            {
               "Text": "Seattle",
               "Score": 0.96,
               "Type": "LOCATION",
               "BeginOffset": 23,
               "EndOffset": 30
            }
         ]
      }
   ],
   "ErrorList": []
}
```

Lorsqu'une erreur se produit dans la demande, la réponse `ErrorList` contient un identifiant les documents contenant une erreur. Le document est identifié par son index dans la liste d'entrée. Par exemple, l'entrée suivante de l'`BatchDetectLanguage`opération contient un document qui ne peut pas être traité :

```
{
   "TextList": [
     "hello friend",
     "$$$$$$",
     "hola amigo"
   ]
}
```

La réponse d'Amazon Comprehend inclut une liste d'erreurs identifiant le document contenant une erreur :

```
{
    "ResultList": [
        {
          "Index": 0,
          "Languages":[
            {
              "LanguageCode":"en",
              "Score": 0.99
            }
          ]
        },
```

```
        {
          "Index": 2
          "Languages":[
            {
              "LanguageCode":"es",
              "Score": 0.82
            }
          ]
        }
    ],
    "ErrorList": [
      {
        "Index": 1,
        "ErrorCode": "InternalServerException",
        "ErrorMessage": "Unexpected Server Error. Please try again."
      }
    ]
}
```

Pour plus d'informations sur les opérations d'API par lots synchrones, consultezLot en temps réel APIs.

Traitement par lots asynchrone

Pour analyser des documents volumineux et de grandes collections de documents, utilisez les opérations asynchrones d'Amazon Comprehend.

Pour analyser un ensemble de documents, vous devez généralement suivre les étapes suivantes :

1. Stockez les documents dans un compartiment Amazon S3.
2. Lancez une ou plusieurs tâches d'analyse pour analyser les documents.
3. Surveillez la progression des tâches d'analyse.
4. Récupérez les résultats de l'analyse depuis un compartiment S3 lorsque le travail est terminé.

Pour plus d'informations sur l'utilisation des opérations d'API asynchrones, consultez Exécution de tâches d'analyse à l'aide de la console (console) et. Tâches d'analyse asynchrones à l'aide de l'API

Langues prises en charge dans Amazon Comprehend

Amazon Comprehend prend en charge une grande variété de langues pour ses différentes fonctionnalités. Les langues prises en charge et les fonctionnalités qui les prennent en charge sont indiquées dans les tableaux suivants.

Rubriques

- Langues prises en charge
- Langues prises en charge par les fonctionnalités d'Amazon Comprehend

Langues prises en charge

Amazon Comprehend (à l'exception de la fonctionnalité de détection de la langue dominante) prend en charge les langues suivantes pour une ou plusieurs fonctionnalités.

Code	Langue
de	Allemand
en	Anglais
es	Espagnol
it	Italien
pt	Portugais
fr	Français
ja	Japonais
ko	Coréen
hi	Hindi
ar	Arabe
zh	Chinois (simplifié)

Code	Langue
zh-TW	Chinois (traditionnel)

> **Note**
>
> Amazon Comprehend identifie la langue à l'aide des identifiants de la RFC 5646. S'il existe un identifiant ISO 639-1 à 2 lettres, avec une sous-étiquette régionale./ Si nécessaire, il l'utilise. Dans le cas contraire, il utilise le code à 3 lettres ISO 639-2.
> Pour plus d'informations sur la RFC 5646, consultez la section Tags pour identifier les langues sur le site Web de l'IETF Tools.

Langues prises en charge par les fonctionnalités d'Amazon Comprehend

Fonctionnalité	Langues prises en charge
Langue dominante	Consultez Langue dominante.
Entités	Toutes les langues prises en charge.
Phrases clés	Toutes les langues prises en charge.
Détection des entités PII	Anglais et espagnol
Étiquetage des entités PII	Anglais et espagnol
Sentiment	Toutes les langues prises en charge.
Sentiment ciblé	Anglais.
Analyse syntaxique	Allemand (de), anglais (en), espagnol (es), français (fr), italien (it) et portugais (pt).
Modélisation des rubriques	Cela ne dépend pas de la langue utilisée. Ne prend pas en charge les langues basées sur

Fonctionnalité	Langues prises en charge
	des caractères telles que le chinois, le japonais et le coréen.
Classification personnalisée	Les modèles en texte brut prennent en charge les langues suivantes : allemand (de), anglais (en), espagnol (es), français (fr), italien (it) et portugais (pt). Les modèles de documents natifs ne prennent en charge que les documents en anglais.
Reconnaissance d'entités personnalisée	Allemand (de), anglais (en), espagnol (es), français (fr), italien (it) et portugais (pt). La reconnaissance d'entités personnalisée pour PDF et Word ne prend en charge que les documents en anglais.

Configuration

Avant d'utiliser Amazon Comprehend pour la première fois, effectuez les tâches suivantes.

Configuration des tâches

- Inscrivez-vous pour un Compte AWS
- Création d’un utilisateur doté d’un accès administratif
- Configurez le AWS Command Line Interface (AWS CLI)
- Octroi d’un accès par programmation

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l’utilisateur racine a accès à l’ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l’utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d’un utilisateur doté d’un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l’aide pour vous connecter en utilisant l’utilisateur racine, consultez Connexion en tant qu’utilisateur racine dans le Guide de l’utilisateur Connexion à AWS .
2. Activez l’authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, consultez la section Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d’un utilisateur doté d’un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d’ AWS IAM Identity Center dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu’utilisateur doté d’un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l’URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l’utilisateur IAM Identity Center.

Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d’un accès à d’autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d’autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Configurez le AWS Command Line Interface (AWS CLI)

Vous n'en avez pas besoin AWS CLI pour effectuer les étapes des exercices de mise en route. Cependant, quelques-uns des autres exercices de ce guide l'exigent. Si vous préférez, vous pouvez ignorer cette étape et passer à Commencer à utiliser Amazon Comprehend la configuration AWS CLI ultérieure.

Pour installer et configurer AWS CLI

1. Installez le AWS CLI. Pour obtenir des instructions, consultez la rubrique suivante du guide de AWS Command Line Interface l'utilisateur :

   Installation ou mise à jour de la dernière version du AWS Command Line Interface
2. Configurez le AWS CLI. Pour obtenir des instructions, consultez la rubrique suivante du guide de AWS Command Line Interface l'utilisateur :

   Configuration de l’interface AWS Command Line Interface (français non garanti)

Octroi d’un accès par programmation

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l’une des options suivantes.

Quel utilisateur a besoin d’un accès programmatique ?	Pour	Par
Identité de la main-d’œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l’interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Configuration du AWS CLI à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, outils, et AWS APIs, voir Authentification IAM Identity Center dans le guide de référence AWS SDKs et Tools.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes	Suivez les instructions de l’interface que vous souhaitez utiliser.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Par
	programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	• Pour le AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'AWS Command Line Interface utilisateur. • Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations d'identif ication à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Commencer à utiliser Amazon Comprehend

L'exercice suivant utilise la console Amazon Comprehend pour créer et exécuter une tâche de détection d'entités asynchrone. Cet exercice suppose que vous connaissez Amazon Simple Storage Service (Amazon S3). Pour un exemple plus simple, voirAnalyse en temps réel à l'aide des modèles intégrés.

Pour créer une tâche de détection d'entités

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Analysis Jobs, puis Create job.
3. Sous Paramètres du job, nommez le job. Le nom doit être unique au sein de la région et du compte.
4. Pour Type d'analyse, sélectionnez Entités.
5. Dans Langue, choisissez la langue des documents d'entrée.
6. Sous Données d'entrée, pour Source de données, sélectionnez Exemples de documents. La console définit l'emplacement S3 comme étant le dossier contenant les échantillons publics.
7. Sous Données de sortie, dans l'emplacement S3, collez l'URL ou l'emplacement du dossier dans Amazon S3 pour les fichiers de sortie.
8. Dans la section Autorisations d'accès, sélectionnez Créer un rôle IAM. La console crée un nouveau rôle IAM avec les autorisations appropriées pour qu'Amazon Comprehend puisse accéder aux compartiments d'entrée et de sortie.
9. Lorsque vous avez fini de remplir le formulaire, choisissez Créer une tâche pour créer et démarrer la tâche de détection de sujets.

   La nouvelle tâche apparaît dans la liste des tâches avec le champ d'état indiquant le statut de la tâche. Le champ peut être `IN_PROGRESS` destiné à une tâche en cours de traitement, `COMPLETED` à une tâche terminée avec succès `FAILED` ou à une tâche comportant une erreur.
10. Choisissez la tâche pour ouvrir le panneau des détails de la tâche.
11. Sous Sortie, dans Emplacement des données de sortie, choisissez le lien pour ouvrir la console Amazon S3.
12. Dans la console Amazon S3, choisissez Télécharger et enregistrez le `output.tar.gz` fichier.
13. Décompressez le fichier et enregistrez-le en tant que fichier Json.

14. Voir the section called “Entités” pour une description des types d'entités et des champs pour chaque entité détectée.

Analyse à l'aide de la console Amazon Comprehend

Vous pouvez utiliser la console Amazon Comprehend pour analyser des documents en temps réel ou pour exécuter des tâches d'analyse asynchrones.

Grâce à l'analyse en temps réel avec des modèles intégrés, vous pouvez reconnaître des entités, extraire des phrases clés, détecter le langage principal, détecter les informations personnelles, déterminer le sentiment, analyser le sentiment ciblé et analyser la syntaxe.

Vous pouvez exécuter des tâches d'analyse à l'aide des modèles intégrés pour trouver des informations telles que des entités, des événements, des phrases, la langue principale, des sentiments, des sentiments ciblés et des informations personnelles identifiables (PII). Vous pouvez également exécuter des tâches de modélisation thématique.

La console prend également en charge les analyses asynchrones et en temps réel à l'aide de modèles personnalisés. Pour plus d'informations, consultez Classification personnalisée et Reconnaissance d'entités personnalisée.

Rubriques

- Analyse en temps réel à l'aide des modèles intégrés
- Exécution de tâches d'analyse à l'aide de la console

Analyse en temps réel à l'aide des modèles intégrés

Vous pouvez utiliser la console Amazon Comprehend pour exécuter une analyse en temps réel d'un document texte codé en UTF-8. Le document peut être en anglais ou dans l'une des autres langues prises en charge par Amazon Comprehend. Les résultats sont affichés dans la console afin que vous puissiez consulter l'analyse.

Pour commencer à analyser des documents, connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.

Vous pouvez remplacer le texte d'exemple par votre propre texte, puis choisir Analyser pour obtenir une analyse de votre texte. Sous le texte en cours d'analyse, le volet Résultats affiche plus d'informations sur le texte.

Exécutez une analyse en temps réel à l'aide du modèle intégré

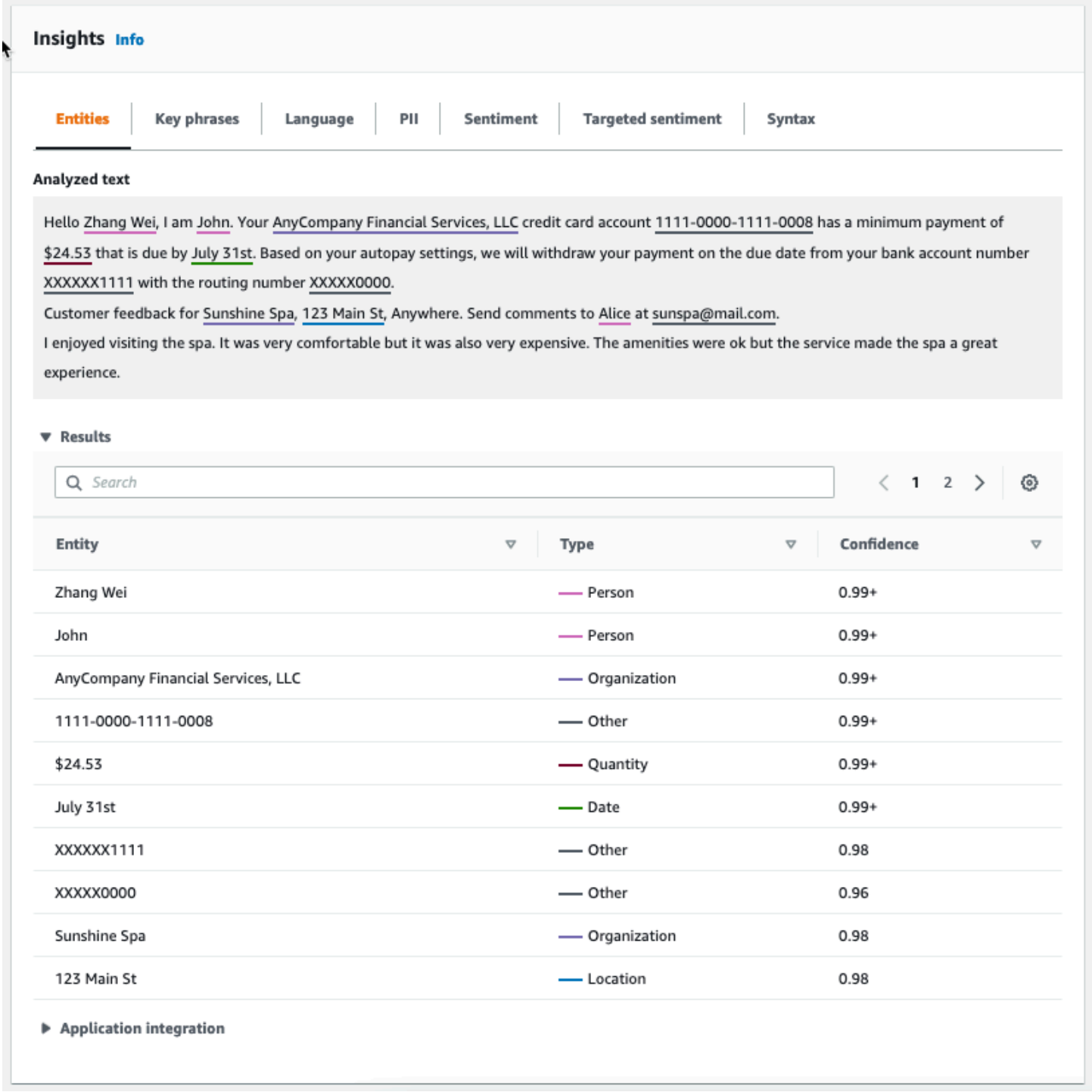
1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, sélectionnez Analyse en temps réel.
3. Sous Type d'entrée, choisissez Intégré pour le type d'analyse.
4. Entrez le texte que vous souhaitez analyser.
5. Choisissez Analyser. La console affiche les résultats de l'analyse de texte dans le panneau Insights. Le panneau Insights inclut un onglet pour chacun des types d'insights. Les sections suivantes décrivent les résultats pour le type d'aperçu.

Rubriques

- Entités
- Phrases clés
- Langue
- données d'identification personnelle (PII)
- Sentiment
- Sentiment ciblé
- Syntaxe

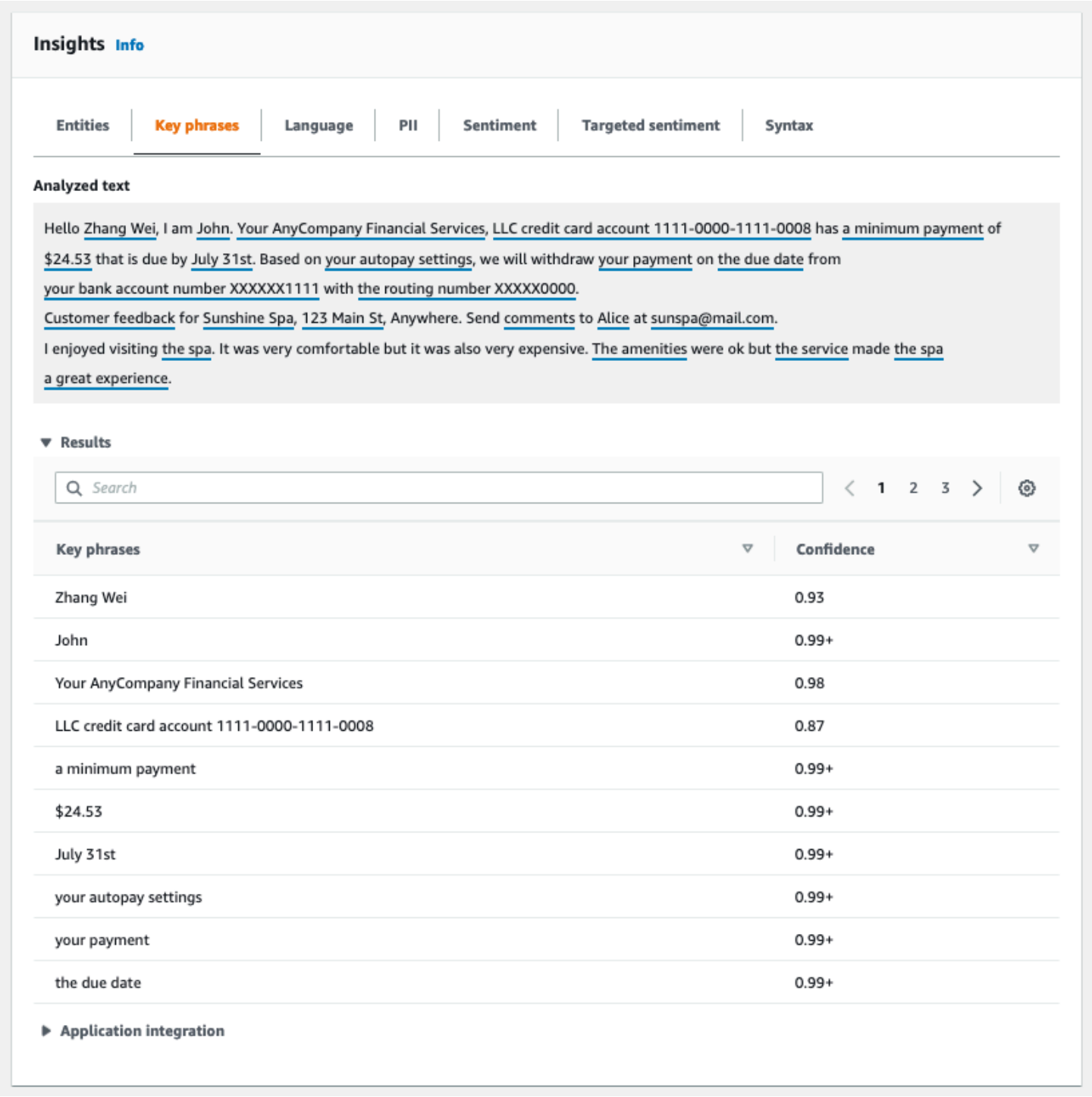
Entités

L'onglet Entités répertorie chaque entité, sa catégorie et le niveau de confiance détecté par Amazon Comprehend dans le texte saisi. Les résultats sont codés par couleur pour indiquer différents types d'entités tels que les organisations, les lieux, les dates et les personnes. Pour de plus amples informations, veuillez consulter Entités.



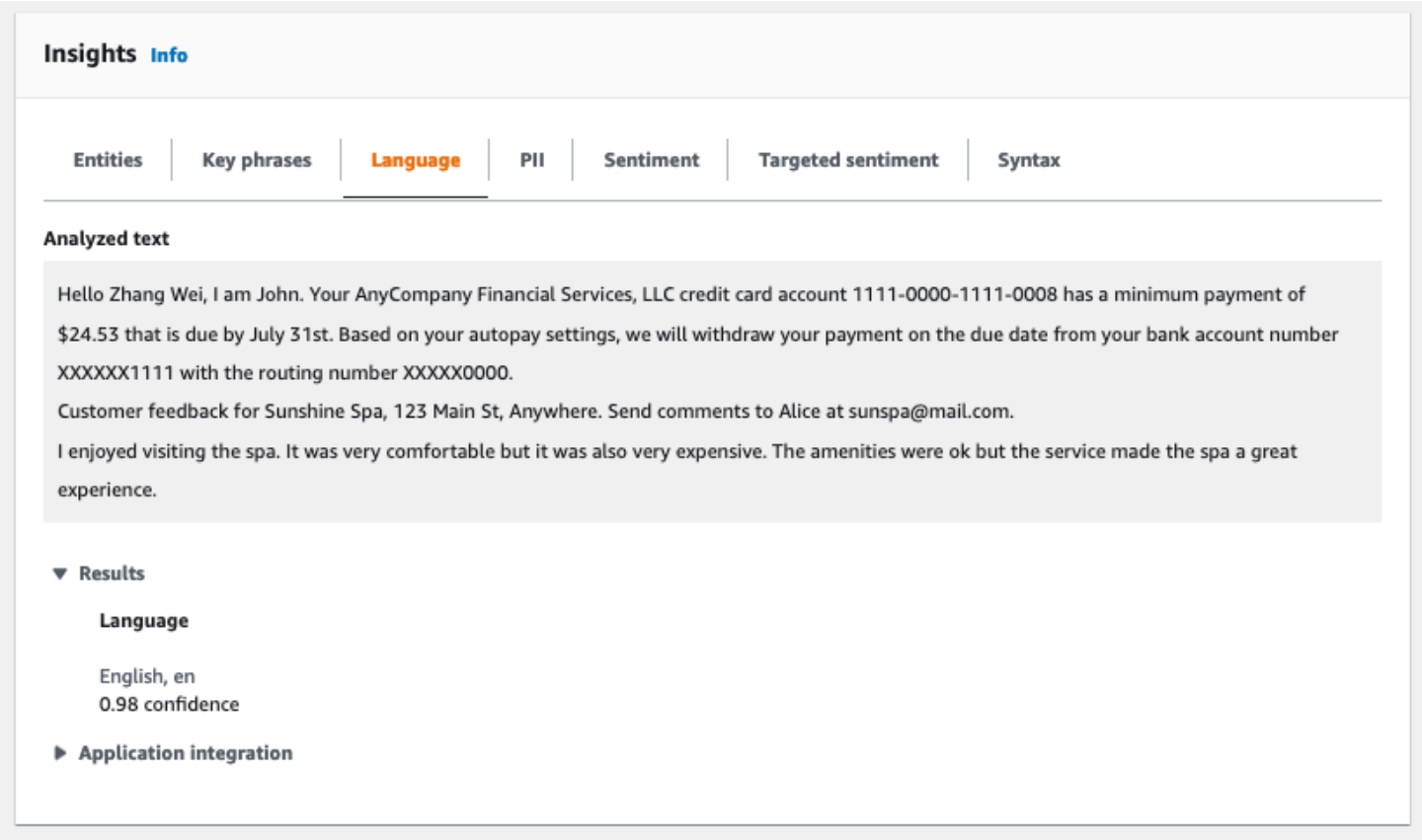
Phrases clés

L'onglet Phrases clés répertorie les phrases nominales clés détectées par Amazon Comprehend dans le texte saisi, ainsi que le niveau de confiance associé. Pour de plus amples informations, veuillez consulter Phrases clés.



Langue

L'onglet Langue indique la langue dominante du texte et le niveau de confiance d'Amazon Comprehend quant à la détection correcte de la langue dominante. Amazon Comprehend peut reconnaître 100 langues. Pour de plus amples informations, veuillez consulter Langue dominante.



données d'identification personnelle (PII)

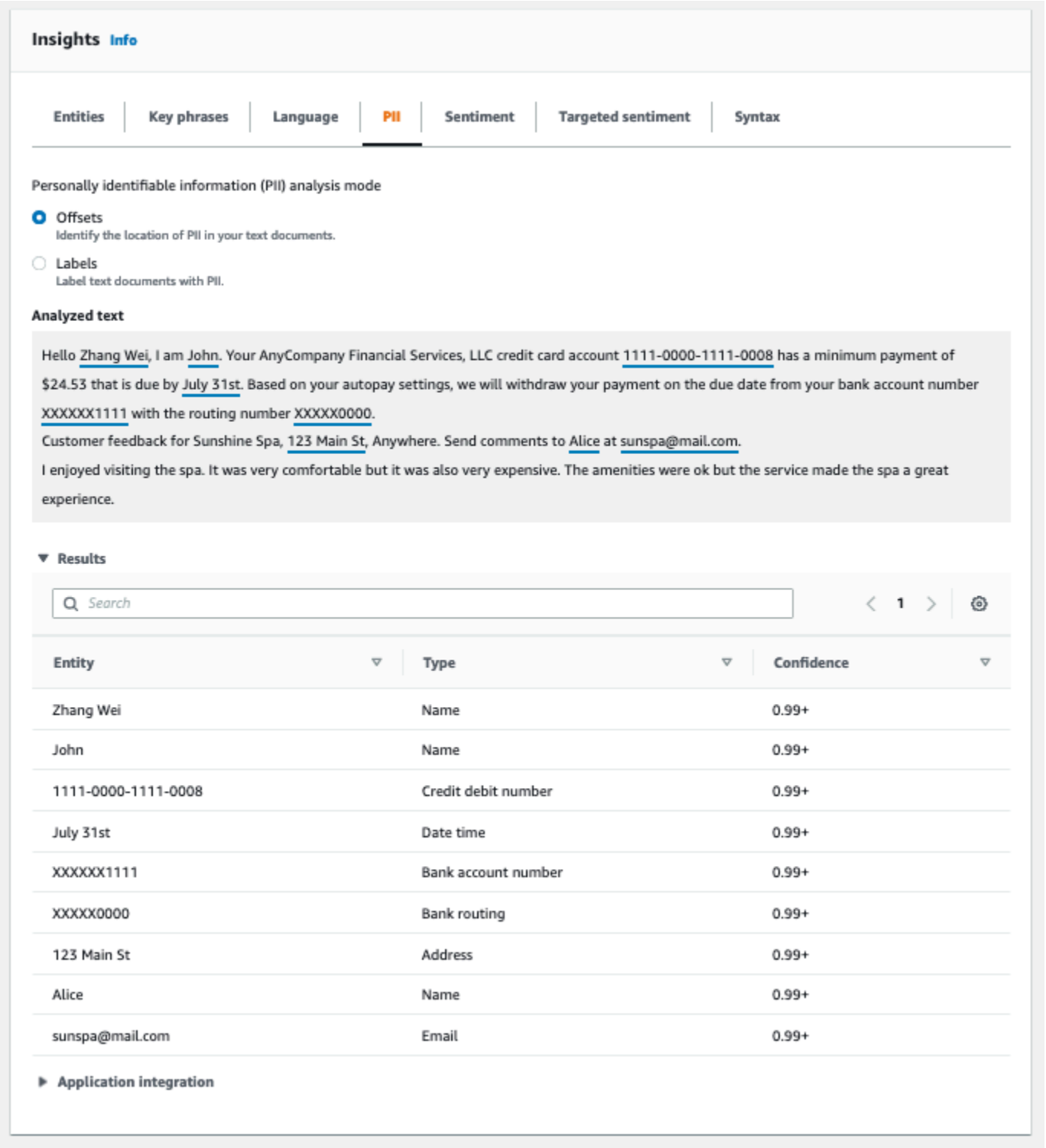
L'onglet PII répertorie les entités dans votre texte saisi qui contiennent des informations personnelles identifiables (PII). Une entité PII est une référence textuelle à des données personnelles qui pourraient être utilisées pour identifier un individu, telles qu'une adresse, un numéro de compte bancaire ou un numéro de téléphone. Pour de plus amples informations, veuillez consulter Détection des entités PII.

L'onglet PII propose deux modes d'analyse :

- Offsets
- Étiquettes

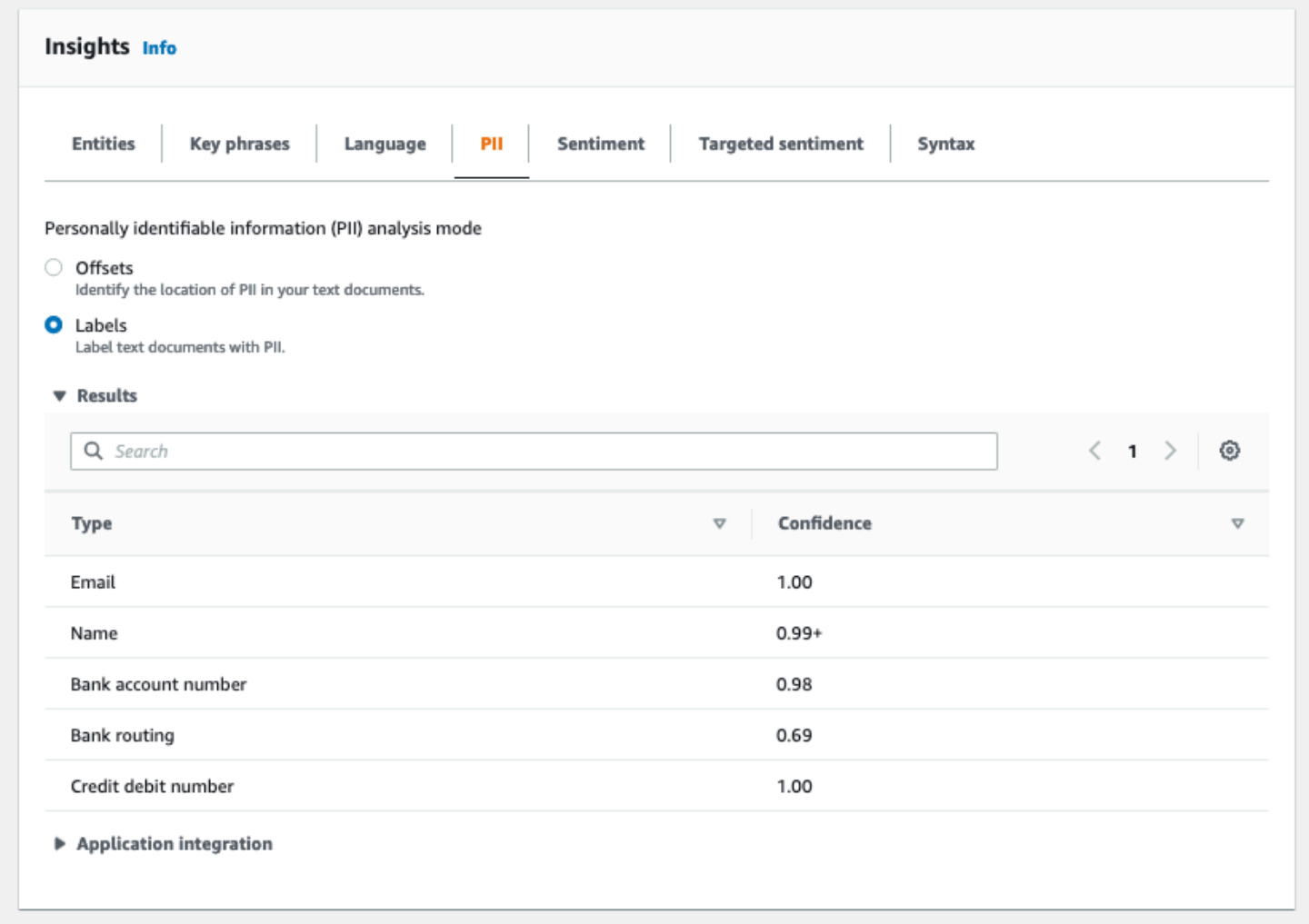
Offsets

Le mode d'analyse des décalages identifie l'emplacement des informations personnelles dans vos documents texte. Pour de plus amples informations, veuillez consulter Localiser les entités PII.


Insights Info
Entities
Key phrases
Language
PII
Sentiment
Targeted sentiment
Syntax
Personally identifiable information (PII) analysis mode
Offsets
Identify the location of PII in your text documents.
Labels
Label text documents with PII.
Analyzed text
Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment of $24.53 that is due by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account number XXXXXX1111 with the routing number XXXXX0000.
Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com.
I enjoyed visiting the spa. It was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great experience.
Results
Search
1
Entity
Type
Confidence
Zhang Wei
Name
0.99+
John
Name
0.99+
1111-0000-1111-0008
Credit debit number
0.99+
July 31st
Date time
0.99+
XXXXXX1111
Bank account number
0.99+
XXXXX0000
Bank routing
0.99+
123 Main St
Address
0.99+
Alice
Name
0.99+
sunspa@mail.com
Email
0.99+
Application integration

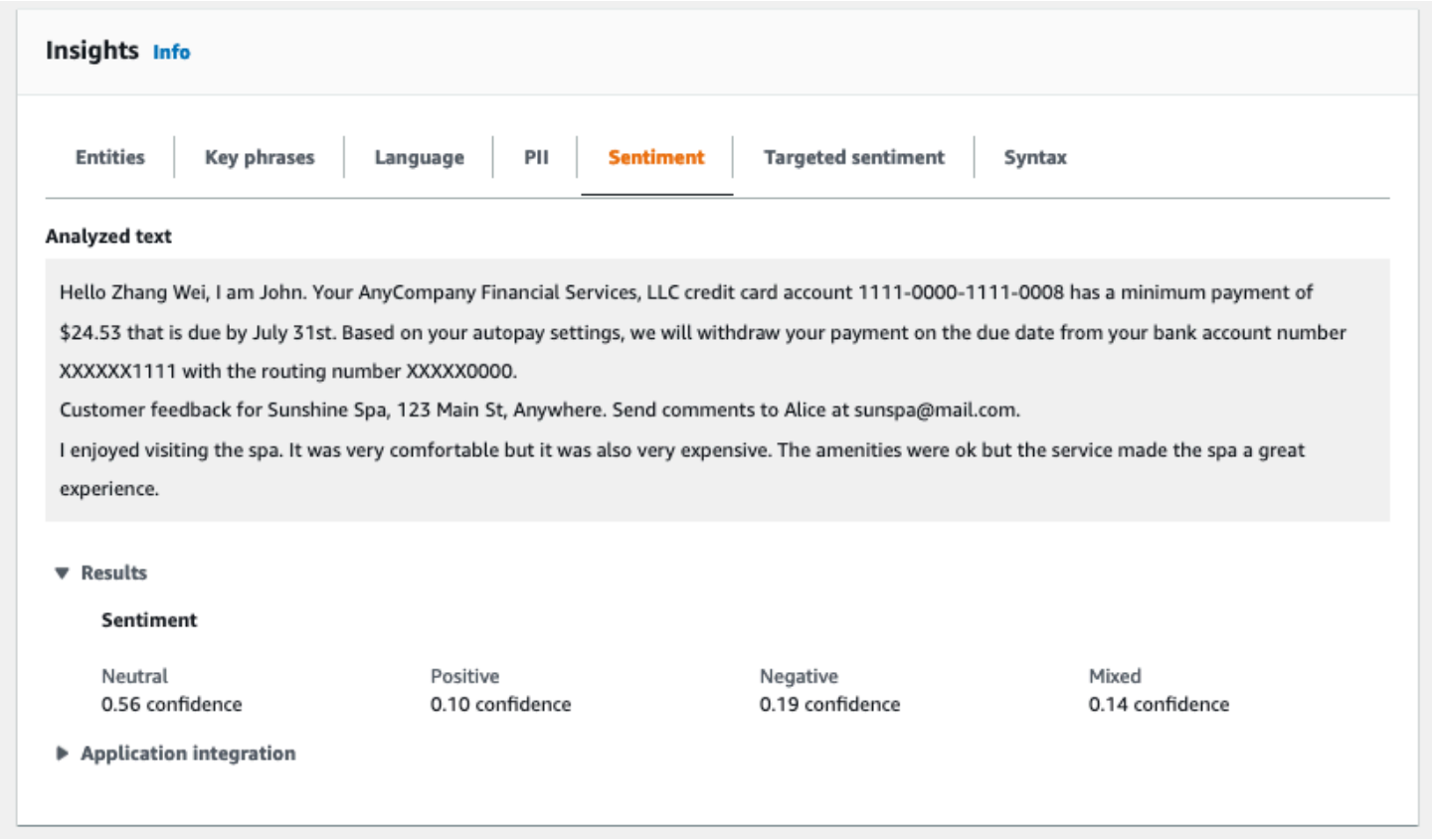
Étiquettes

Le mode d'analyse des étiquettes vérifie la présence de PII dans votre document texte et renvoie les étiquettes des types d'entités PII identifiés. Pour de plus amples informations, veuillez consulter Étiquetage des entités PII.



Sentiment

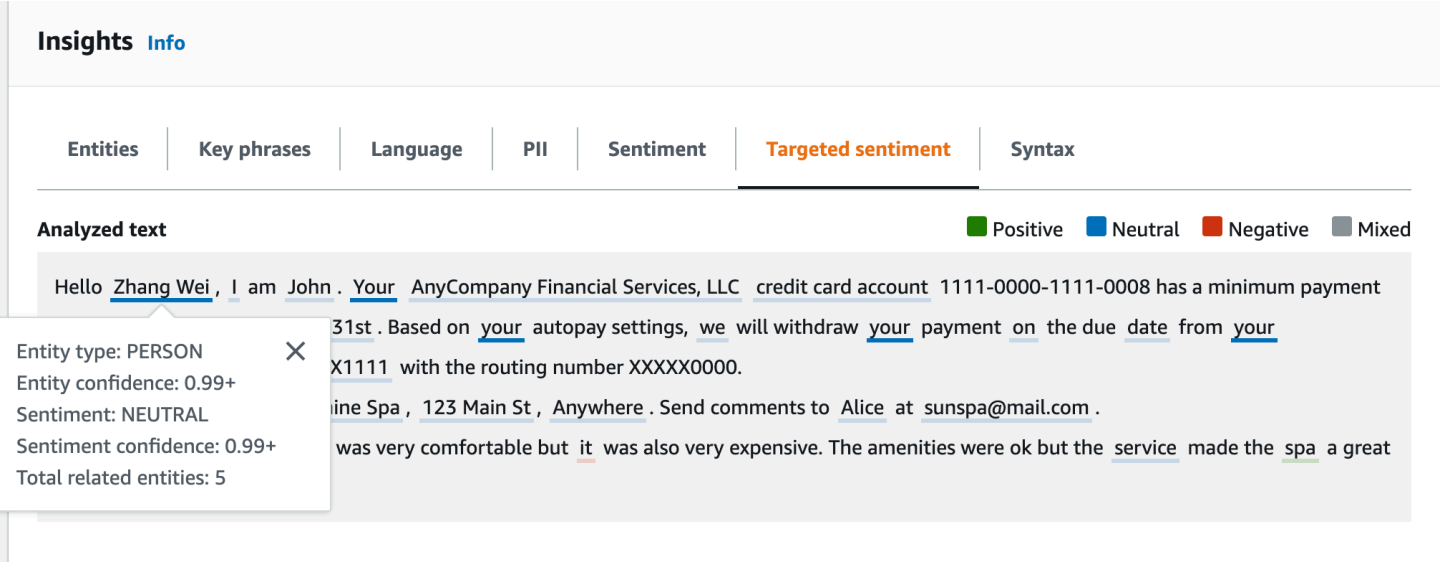
L'onglet Sentiment indique le sentiment dominant du texte. Le sentiment peut être évalué comme neutre, positif, négatif ou mitigé. Dans ce cas, chaque sentiment possède un indice de confiance, fournissant une estimation par Amazon Comprehend indiquant que ce sentiment est dominant. Pour de plus amples informations, veuillez consulter Sentiment.



Sentiment ciblé

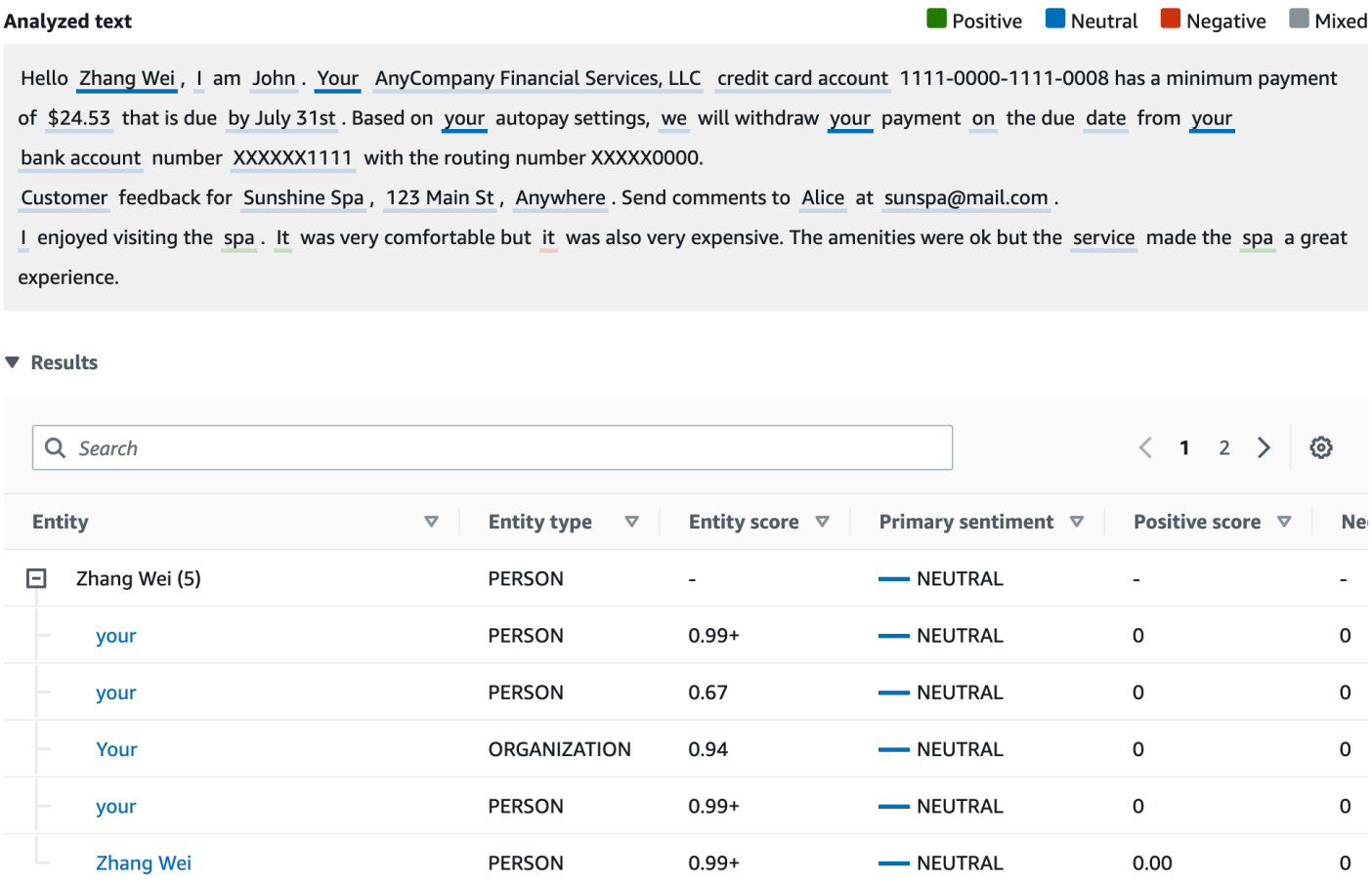
L'analyse ciblée des sentiments identifie les sentiments exprimés à l'égard des entités mentionnées dans le texte. Amazon Comprehend attribue une note de sentiment à chaque mention d'une entité, ainsi qu'une note de confiance et d'autres informations. Une évaluation des sentiments peut être neutre, positive, négative ou mitigée.

Dans le panneau Texte analysé, la console souligne chacune des entités analysées. La couleur du texte souligné indique le sentiment général de l'entité. Si vous passez le curseur sur une entité, la console affiche des informations supplémentaires dans une fenêtre contextuelle.



Le tableau des résultats fournit des informations supplémentaires sur chaque entité. S'il existe plusieurs mentions de la même entité, appelées groupe de coréférence, le tableau affiche ces mentions sous la forme d'un ensemble démontable de lignes associées à l'entité principale.

Dans l'exemple suivant, l'entité est une personne nommée Zhang Wei. L'analyse ciblée des sentiments reconnaît que chaque mention de vous fait référence à la même personne. La console affiche ces mentions sous forme de sous-entrées de l'entité principale.

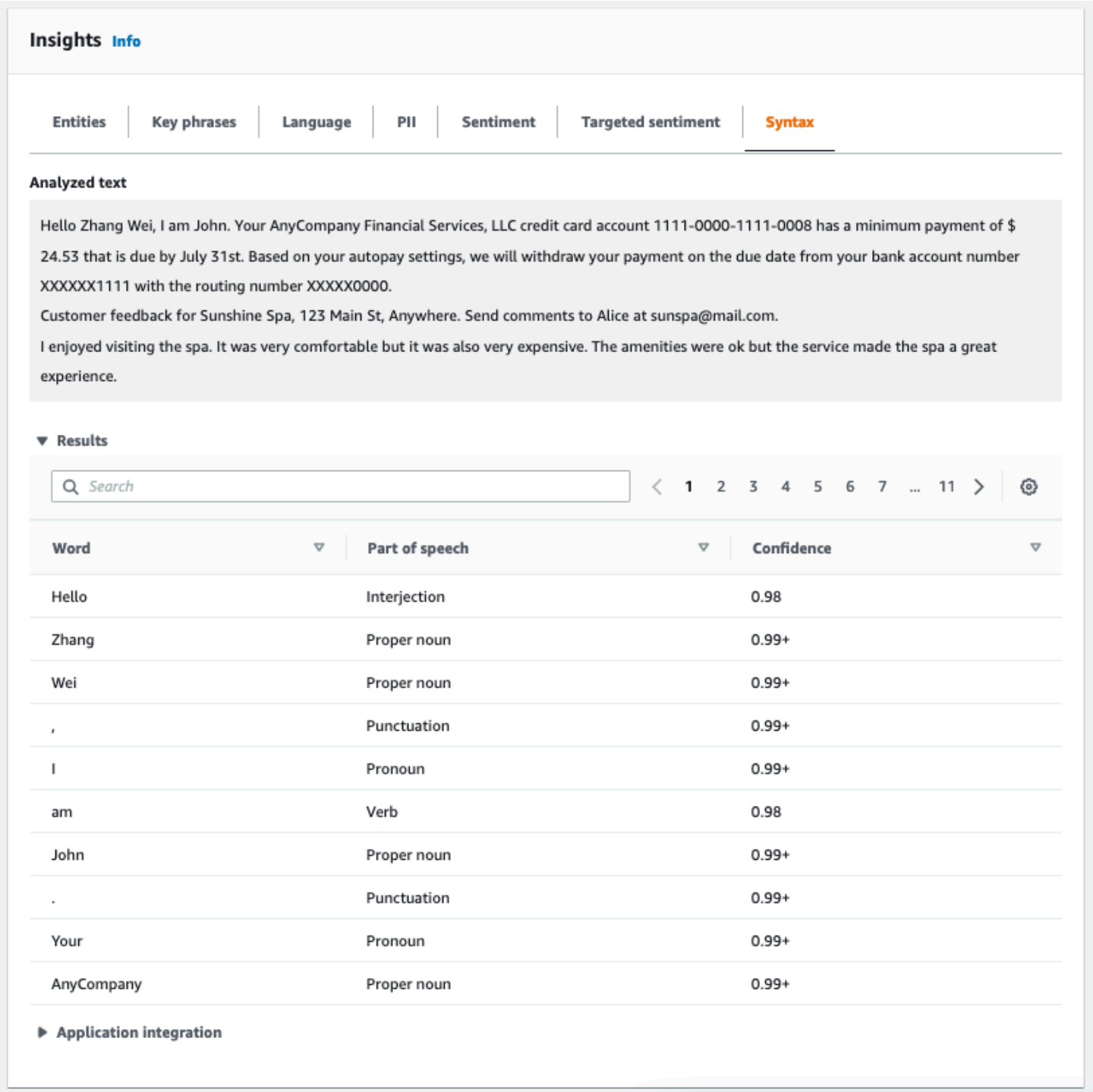


Si le texte que vous analysez n'inclut aucun sentiment ciblé[Types d'entités](), l'analyse du sentiment ciblé affiche un champ de résultats vide.

Pour plus d'informations sur l'utilisation de la console pour une analyse ciblée des sentiments en temps réel, consultez[Analyse en temps réel à l'aide de la console]().

Syntaxe

L'onglet Syntaxe affiche une ventilation de chaque élément du texte, ainsi que sa partie du discours et le score de confiance associé. Pour de plus amples informations, veuillez consulter [Analyse syntaxique]().



Exécution de tâches d'analyse à l'aide de la console

Vous pouvez utiliser la console Amazon Comprehend pour créer et gérer des tâches d'analyse asynchrones. Votre travail analyse les documents stockés dans Amazon S3 pour trouver des entités

telles que des événements, des phrases, la langue principale, des sentiments ou des informations personnelles identifiables (PII).

Pour créer une tâche d'analyse

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Tâches d'analyse, puis sélectionnez Créer une tâche.
3. Sous Paramètres de la tâche, attribuez un nom unique à la tâche d'analyse.
4. Pour Type d'analyse, choisissez l'un des types d'analyse intégrés.

   Si vous choisissez le langage principal ou la modélisation par sujet, vous pouvez ignorer l'étape suivante.
5. Selon le type d'analyse que vous choisissez, la console affiche un ou plusieurs des champs supplémentaires suivants :

   - La langue est requise pour tous les types d'analyse intégrés, à l'exception du langage principal et de la modélisation thématique.

     Choisissez la langue de vos documents d'entrée.
   - Les types d'événements cibles sont obligatoires pour le type d'analyse des événements.

     Sélectionnez les types d'événements à détecter dans vos documents d'entrée. Pour plus d'informations sur les types d'événements pris en charge, consultezTypes d'événements.
   - Les paramètres de détection des informations personnelles sont requis pour le type d'analyse des informations personnelles.

     Sélectionnez le mode de sortie. Pour plus d'informations sur les paramètres de détection des informations personnelles, consultezDétection des entités PII.
6. Sous Données d'entrée, spécifiez l'emplacement des documents d'entrée dans Amazon S3 :
   - Pour analyser vos propres documents, choisissez Mes documents, puis Browse S3 pour indiquer le chemin d'accès au compartiment ou au dossier contenant vos fichiers.
   - Pour analyser les échantillons fournis par Amazon Comprehend, sélectionnez Example documents. Dans ce cas, Amazon Comprehend utilise un bucket géré par AWS, et vous n'en spécifiez pas l'emplacement.
7. (Facultatif) Dans Format d'entrée, spécifiez l'un des formats suivants pour vos fichiers d'entrée :

- Un document par fichier — Chaque fichier contient un document d'entrée. C'est la meilleure solution pour les collections de documents volumineux.
- Un document par ligne — L'entrée est un ou plusieurs fichiers. Chaque ligne d'un fichier est considérée comme un document. C'est la meilleure solution pour les documents courts, tels que les publications sur les réseaux sociaux. Chaque ligne doit se terminer par une ligne d'alimentation (LF,\n), un retour en chariot (CR, \ r), ou les deux (CRLF, \ r\n). Vous ne pouvez pas utiliser le séparateur de ligne UTF-8 (u+2028) pour terminer une ligne.

8. Sous Données de sortie, choisissez Browse S3. Choisissez le compartiment ou le dossier Amazon S3 dans lequel vous souhaitez qu'Amazon Comprehend écrive les données de sortie produites par l'analyse.
9. (Facultatif) Pour chiffrer le résultat de sortie de votre tâche, choisissez Chiffrement. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte :
   - Si vous utilisez une clé associée au compte actuel, choisissez l'alias de clé ou l'ID pour l'ID de clé KMS.
   - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'alias ou de l'ID de clé sous ID de clé KMS.

   > **Note**
   >
   > Pour plus d'informations sur la création et l'utilisation des clés KMS et le chiffrement associé, consultez la section Service de gestion des clés (KMS).

10. Sous Autorisations d'accès, fournissez un rôle IAM qui :
    - Accorde un accès en lecture à l'emplacement Amazon S3 de vos documents d'entrée.
    - Accorde un accès en écriture à l'emplacement Amazon S3 de vos documents de sortie.
    - Inclut une politique de confiance qui permet au directeur du `comprehend.amazonaws.com` service d'assumer le rôle et d'obtenir ses autorisations.

    Si vous ne possédez pas encore de rôle IAM doté de ces autorisations et d'une politique de confiance appropriée, choisissez Créer un rôle IAM pour en créer un.
11. Lorsque vous avez fini de remplir le formulaire, choisissez Créer une tâche pour créer et démarrer la tâche de détection de sujets.

La nouvelle tâche apparaît dans la liste des tâches avec le champ d'état indiquant le statut de la tâche. Le champ peut être `IN_PROGRESS` destiné à une tâche en cours de traitement, `COMPLETED` à une tâche terminée avec succès `FAILED` ou à une tâche comportant une erreur. Vous pouvez cliquer sur une tâche pour obtenir plus d'informations sur la tâche, y compris les messages d'erreur éventuels.

Lorsque la tâche est terminée, Amazon Comprehend stocke les résultats de l'analyse dans l'emplacement Amazon S3 en sortie que vous avez spécifié pour la tâche. Pour une description des résultats de l'analyse pour chaque type d'aperçu, voirInformations.

Utilisation de l'API Amazon Comprehend

L'API Amazon Comprehend prend en charge les opérations d'analyse en temps réel (synchrone) et les opérations de démarrage et de gestion des tâches d'analyse asynchrones.

Vous pouvez utiliser directement les opérateurs d'API Amazon Comprehend, ou vous pouvez utiliser la CLI ou l'un des. SDKs Les exemples présentés dans ce chapitre utilisent la CLI, le SDK Python et le SDK Java.

Pour exécuter les exemples AWS CLI et Python, vous devez installer le AWS CLI. Pour de plus amples informations, veuillez consulter Configurez le AWS Command Line Interface (AWS CLI).

Pour exécuter les exemples Java, vous devez installer le AWS SDK pour Java. Pour obtenir des instructions relatives à l'installation du SDK pour Java, consultez la section Configurer AWS le SDK pour Java.

Rubriques

- Utilisation d'Amazon Comprehend avec un SDK AWS
- Analyse en temps réel à l'aide de l'API
- Tâches d'analyse asynchrones à l'aide de l'API

Utilisation d'Amazon Comprehend avec un SDK AWS

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code

Documentation SDK	Exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Analyse en temps réel à l'aide de l'API

Les exemples suivants montrent comment utiliser l'API Amazon Comprehend pour une analyse en temps réel AWS CLI, en utilisant et AWS SDKs pour .NET, Java et Python. Utilisez les exemples pour en savoir plus sur les opérations synchrones d'Amazon Comprehend et comme éléments de base pour vos propres applications.

Les exemples .NET présentés dans cette section utilisent le kit AWS SDK pour .NET. Vous pouvez utiliser le AWS Toolkit for Visual Studiopour développer des AWS applications à l'aide de .NET. Il inclut des modèles utiles et l' AWS explorateur pour le déploiement d'applications et la gestion des services. Pour connaître le point de vue des développeurs .NET AWS, consultez le AWS guide destiné aux développeurs .NET.

Rubriques

- Détecter la langue dominante
- Détection des entités nommées
- Détecter les phrases clés
- Déterminer le sentiment
- Analyse en temps réel pour un sentiment ciblé
- Détection de la syntaxe
- Lot en temps réel APIs

Détecter la langue dominante

Pour déterminer la langue dominante utilisée dans le texte, utilisez l'DetectDominantLanguageopération. Pour détecter la langue dominante dans un maximum de 25 documents par lot, utilisez l'BatchDetectDominantLanguageopération. Pour de plus amples informations, veuillez consulter Lot en temps réel APIs.

Rubriques

- À l'aide du AWS Command Line Interface
- À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

À l'aide du AWS Command Line Interface

L'exemple suivant montre comment utiliser l'`DetectDominantLanguage`opération avec le AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-dominant-language \
    --region region \
    --text "It is raining today in Seattle."
```

Amazon Comprehend répond comme suit :

```
{
```

```
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9793661236763
        }
    ]
}
```

À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

Pour des exemples de SDK expliquant comment déterminer la langue dominante, consultezUtilisation `DetectDominantLanguage` avec un AWS SDK ou une CLI.

Détection des entités nommées

Pour déterminer les entités nommées dans un document, utilisez l'DetectEntitiesopération. Pour détecter des entités dans un maximum de 25 documents par lot, utilisez l'BatchDetectEntitiesopération. Pour de plus amples informations, veuillez consulter Lot en temps réel APIs.

Rubriques

- À l'aide du AWS Command Line Interface
- À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

À l'aide du AWS Command Line Interface

L'exemple suivant illustre l'utilisation de l'`DetectEntities`opération à l'aide du AWS CLI. Vous devez spécifier la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-entities \
    --region region \
    --language-code "en" \
    --text "It is raining today in Seattle."
```

Amazon Comprehend répond comme suit :

```
{
    "Entities": [
        {
            "Text": "today",
            "Score": 0.97,
            "Type": "DATE",
            "BeginOffset": 14,
            "EndOffset": 19
        },
        {
            "Text": "Seattle",
            "Score": 0.95,
            "Type": "LOCATION",
            "BeginOffset": 23,
            "EndOffset": 30
        }
    ],
    "LanguageCode": "en"
}
```


À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

Pour des exemples de SDK expliquant comment déterminer la langue dominante, consultezUtilisation `DetectEntities` avec un AWS SDK ou une CLI.

Détecter les phrases clés

Pour déterminer les phrases nominales clés utilisées dans le texte, utilisez l'DetectKeyPhrasesopération. Pour détecter les phrases nominales clés dans un maximum de 25 documents par lot, utilisez cette BatchDetectKeyPhrasesopération. Pour de plus amples informations, veuillez consulter Lot en temps réel APIs.

Rubriques

- À l'aide du AWS Command Line Interface
- À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

À l'aide du AWS Command Line Interface

L'exemple suivant montre comment utiliser l'`DetectKeyPhrases`opération avec le AWS CLI. Vous devez spécifier la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-key-phrases \
    --region region \
    --language-code "en" \
    --text "It is raining today in Seattle."
```

Amazon Comprehend répond comme suit :

```
{
    "LanguageCode": "en",
    "KeyPhrases": [
        {
            "Text": "today",
            "Score": 0.89,
            "BeginOffset": 14,
            "EndOffset": 19
        },
        {
            "Text": "Seattle",
            "Score": 0.91,
            "BeginOffset": 23,
            "EndOffset": 30
        }
    ]
}
```

À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

Pour des exemples de SDK qui détectent des phrases clés, consultezUtilisation `DetectKeyPhrases` avec un AWS SDK ou une CLI.

Déterminer le sentiment

Amazon Comprehend fournit les opérations d'API suivantes pour analyser les sentiments :

- DetectSentiment— Détermine le sentiment émotionnel général d'un document.
- BatchDetectSentiment— Déterminez le sentiment général d'un maximum de 25 documents par lot. Pour plus d'informations, consultez Lot en temps réel APIs.

- StartSentimentDetectionJob— Démarre une tâche de détection des sentiments asynchrone pour un ensemble de documents.
- ListSentimentDetectionJobs— Renvoie la liste des tâches de détection des sentiments que vous avez soumises.
- DescribeSentimentDetectionJob— Obtient les propriétés (y compris le statut) associées à la tâche de détection des sentiments spécifiée.
- StopSentimentDetectionJob— Arrête la tâche d'évaluation des sentiments en cours spécifiée.

Rubriques

- À l'aide du AWS Command Line Interface
- À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

À l'aide du AWS Command Line Interface

L'exemple suivant montre comment utiliser l'`DetectSentiment`opération avec le AWS CLI. Cet exemple indique la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-sentiment \
    --region region \
    --language-code "en" \
    --text "It is raining today in Seattle."
```

Amazon Comprehend répond comme suit :

```
{
    "SentimentScore": {
        "Mixed": 0.014585512690246105,
        "Positive": 0.31592071056365967,
        "Neutral": 0.5985543131828308,
        "Negative": 0.07093945890665054
    },
    "Sentiment": "NEUTRAL",
    "LanguageCode": "en"
}
```

À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

Pour des exemples de SDK qui déterminent le sentiment du texte saisi, voirUtilisation `DetectSentiment` avec un AWS SDK ou une CLI.

Analyse en temps réel pour un sentiment ciblé

Amazon Comprehend fournit les opérations d'API suivantes pour une analyse ciblée des sentiments en temps réel :

- DetectTargetedSentiment— Analyse le sentiment des entités mentionnées dans un document.
- BatchDetectTargetedSentiment— Analyse le sentiment ciblé pour un maximum de 25 documents par lot. Pour plus d'informations, consultez Lot en temps réel APIs.

Si le texte que vous analysez n'inclut aucun sentiment ciblé Types d'entités, l'API renvoie un tableau d'entités vide.

À l'aide du AWS Command Line Interface

L'exemple suivant montre comment utiliser l'`DetectTargetedSentiment`opération avec le AWS CLI. Cet exemple indique la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-targeted-sentiment \
    --region region \
    --language-code "en" \
    --text "The burger was cooked perfectly but it was cold. The service was OK."
```

Amazon Comprehend répond comme suit :

```
{
"Entities": [
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
```

```
          "BeginOffset": 4,
          "EndOffset": 10,
          "Score": 1,
          "GroupScore": 1,
          "Text": "burger",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "POSITIVE",
            "SentimentScore": {
              "Mixed": 0.001515,
              "Negative": 0.000822,
              "Neutral": 0.000243,
              "Positive": 0.99742
            }
          }
        },
        {
          "BeginOffset": 36,
          "EndOffset": 38,
          "Score": 0.999843,
          "GroupScore": 0.999661,
          "Text": "it",
          "Type": "OTHER",
          "MentionSentiment": {
            "Sentiment": "NEGATIVE",
            "SentimentScore": {
              "Mixed": 0,
              "Negative": 0.999996,
              "Neutral": 0.000004,
              "Positive": 0
            }
          }
        }
      ]
    },
    {
      "DescriptiveMentionIndex": [
        0
      ],
      "Mentions": [
        {
          "BeginOffset": 53,
          "EndOffset": 60,
          "Score": 1,
```

```
          "GroupScore": 1,
          "Text": "service",
          "Type": "ATTRIBUTE",
          "MentionSentiment": {
            "Sentiment": "NEUTRAL",
            "SentimentScore": {
              "Mixed": 0.000033,
              "Negative": 0.000089,
              "Neutral": 0.993325,
              "Positive": 0.006553
            }
          }
        }
      ]
    }
  ]
}
```

Détection de la syntaxe

Pour analyser le texte afin d'extraire les mots individuels et de déterminer les parties du discours pour chaque mot, utilisez l'DetectSyntaxopération. Pour analyser la syntaxe d'un maximum de 25 documents par lot, utilisez l'BatchDetectSyntaxopération. Pour de plus amples informations, veuillez consulter Lot en temps réel APIs.

Rubriques

- En utilisant le AWS Command Line Interface.
- À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

En utilisant le AWS Command Line Interface.

L'exemple suivant montre comment utiliser l'`DetectSyntax`opération avec le AWS CLI. Cet exemple indique la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-syntax \
   --region region \
```

```
    --language-code "en" \
    --text "It is raining today in Seattle."
```

Amazon Comprehend répond comme suit :


```
{
    "SyntaxTokens": [
        {
            "Text": "It",
            "EndOffset": 2,
            "BeginOffset": 0,
            "PartOfSpeech": {
                "Tag": "PRON",
                "Score": 0.8389829397201538
            },
            "TokenId": 1
        },
        {
            "Text": "is",
            "EndOffset": 5,
            "BeginOffset": 3,
            "PartOfSpeech": {
                "Tag": "AUX",
                "Score": 0.9189288020133972
            },
            "TokenId": 2
        },
        {
            "Text": "raining",
            "EndOffset": 13,
            "BeginOffset": 6,
            "PartOfSpeech": {
                "Tag": "VERB",
                "Score": 0.9977611303329468
            },
            "TokenId": 3
        },
        {
            "Text": "today",
            "EndOffset": 19,
            "BeginOffset": 14,
            "PartOfSpeech": {
                "Tag": "NOUN",
```

```
                "Score": 0.9993606209754944
            },
            "TokenId": 4
        },
        {
            "Text": "in",
            "EndOffset": 22,
            "BeginOffset": 20,
            "PartOfSpeech": {
                "Tag": "ADP",
                "Score": 0.9999061822891235
            },
            "TokenId": 5
        },
        {
            "Text": "Seattle",
            "EndOffset": 30,
            "BeginOffset": 23,
            "PartOfSpeech": {
                "Tag": "PROPN",
                "Score": 0.9940338730812073
            },
            "TokenId": 6
        },
        {
            "Text": ".",
            "EndOffset": 31,
            "BeginOffset": 30,
            "PartOfSpeech": {
                "Tag": "PUNCT",
                "Score": 0.9999997615814209
            },
            "TokenId": 7
        }
    ]
}
```


À l'aide du AWS SDK pour Java SDK pour Python, ou SDK pour .NET

Pour des exemples de SDK qui détectent la syntaxe du texte saisi, consultezUtilisation `DetectSyntax` avec un AWS SDK ou une CLI.

Lot en temps réel APIs

Pour envoyer des lots contenant jusqu'à 25 documents, vous pouvez utiliser les opérations par lots en temps réel d'Amazon Comprehend. L'appel d'une opération par lots est identique à l'appel d'un seul document APIs pour chaque document de la demande. L'utilisation du lot APIs peut améliorer les performances de vos applications. Pour de plus amples informations, veuillez consulter Traitement synchrone de plusieurs documents.

Rubriques

- Traitement par lots avec AWS CLI
- Traitement par lots avec AWS SDK pour .NET

Traitement par lots avec AWS CLI

Ces exemples montrent comment utiliser les opérations d'API par lots à l'aide du AWS Command Line Interface. Toutes les opérations sauf l'`BatchDetectDominantLanguage`utilisation du fichier JSON suivant `process.json` appelé en entrée. Pour cette opération, l'`LanguageCode`entité n'est pas incluse.

Le troisième document du fichier JSON (`"$$$$$$$$"`) provoquera une erreur lors du traitement par lots. Il est inclus afin que les opérations incluent un BatchItemErrordans la réponse.

```
{
   "LanguageCode": "en",
   "TextList": [
      "I have been living in Seattle for almost 4 years",
      "It is raining today in Seattle",
      "$$$$$$$$"
   ]
}
```

Les exemples sont formatés pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Rubriques

- Détecter la langue dominante à l'aide d'un batch (AWS CLI)
- Détecter les entités à l'aide d'un lot (AWS CLI)

- Détecter les phrases clés à l'aide d'un batch (AWS CLI)
- Détectez les sentiments à l'aide d'un batch (AWS CLI)

Détecter la langue dominante à l'aide d'un batch (AWS CLI)

L'BatchDetectDominantLanguageopération détermine la langue dominante de chaque document d'un lot. Pour obtenir la liste des langues qu'Amazon Comprehend peut détecter, consultez. Langue dominante La AWS CLI commande suivante lance l'`BatchDetectDominantLanguage`opération.

```
aws comprehend batch-detect-dominant-language \
    --endpoint endpoint \
    --region region \
    --cli-input-json file://path to input file/process.json
```

Voici la réponse de l'`BatchDetectDominantLanguage`opération :

```
{
    "ResultList": [
        {
          "Index": 0,
          "Languages":[
            {
              "LanguageCode":"en",
              "Score": 0.99
            }
          ]
        },
        {
          "Index": 1
          "Languages":[
            {
              "LanguageCode":"en",
              "Score": 0.82
            }
          ]
        }
    ],
    "ErrorList": [
      {
        "Index": 2,
        "ErrorCode": "InternalServerException",
        "ErrorMessage": "Unexpected Server Error. Please try again."
```

```
      }
    ]
}
```

Détecter les entités à l'aide d'un lot (AWS CLI)

Utilisez cette BatchDetectEntitiesopération pour rechercher les entités présentes dans un lot de documents. Pour plus d'informations sur les entités, consultez Entités. La AWS CLI commande suivante lance l'`BatchDetectEntities`opération.

```
aws comprehend batch-detect-entities \
    --endpoint endpoint \
    --region region \
    --cli-input-json file://path to input file/process.json
```

Détecter les phrases clés à l'aide d'un batch (AWS CLI)

L'BatchDetectKeyPhrasesopération renvoie les phrases nominales clés d'un lot de documents. La AWS CLI commande suivante lance l'`BatchDetectKeyNounPhrases`opération.

```
aws comprehend batch-detect-key-phrases
    --endpoint endpoint
    --region region
    --cli-input-json file://path to input file/process.json
```

Détectez les sentiments à l'aide d'un batch (AWS CLI)

Détectez l'impression générale d'un lot de documents à l'aide de cette BatchDetectSentimentopération. La AWS CLI commande suivante lance l'`BatchDetectSentiment`opération.

```
aws comprehend batch-detect-sentiment \
    --endpoint endpoint \
    --region region \
    --cli-input-json file://path to input file/process.json
```

Traitement par lots avec AWS SDK pour .NET

L'exemple de programme suivant montre comment utiliser l'BatchDetectEntitiesopération avec le SDK pour .NET. La réponse du serveur contient un BatchDetectEntitiesItemResultobjet pour chaque document traité avec succès. En cas d'erreur lors du traitement d'un document, un enregistrement

figurera dans la liste des erreurs de la réponse. L'exemple récupère chacun des documents contenant une erreur et les renvoie.

L'exemple .NET présenté dans cette section utilise le AWS SDK pour .NET. Vous pouvez utiliser le AWS Toolkit for Visual Studiopour développer des AWS applications à l'aide de .NET. Il inclut des modèles utiles et l' AWS explorateur pour le déploiement d'applications et la gestion des services. Pour connaître le point de vue des développeurs .NET AWS, consultez le AWS guide destiné aux développeurs .NET.

```
using System;
using System.Collections.Generic;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

namespace Comprehend
{
    class Program
    {
        // Helper method for printing properties
        static private void PrintEntity(Entity entity)
        {
            Console.WriteLine("     Text: {0}, Type: {1}, Score: {2}, BeginOffset: {3}
 EndOffset: {4}",
                entity.Text, entity.Type, entity.Score, entity.BeginOffset,
 entity.EndOffset);
        }

        static void Main(string[] args)
        {
            AmazonComprehendClient comprehendClient = new
 AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

            List<String> textList = new List<String>()
            {
                { "I love Seattle" },
                { "Today is Sunday" },
                { "Tomorrow is Monday" },
                { "I love Seattle" }
            };

            // Call detectEntities API
            Console.WriteLine("Calling BatchDetectEntities");
```

```
            BatchDetectEntitiesRequest batchDetectEntitiesRequest = new
BatchDetectEntitiesRequest()
            {
                TextList = textList,
                LanguageCode = "en"
            };
            BatchDetectEntitiesResponse batchDetectEntitiesResponse =
comprehendClient.BatchDetectEntities(batchDetectEntitiesRequest);

            foreach (BatchDetectEntitiesItemResult item in
batchDetectEntitiesResponse.ResultList)
            {
                Console.WriteLine("Entities in {0}:", textList[item.Index]);
                foreach (Entity entity in item.Entities)
                    PrintEntity(entity);
            }

            // check if we need to retry failed requests
            if (batchDetectEntitiesResponse.ErrorList.Count != 0)
            {
                Console.WriteLine("Retrying Failed Requests");
                List<String> textToRetry = new List<String>();
                foreach(BatchItemError errorItem in
batchDetectEntitiesResponse.ErrorList)
                    textToRetry.Add(textList[errorItem.Index]);

                batchDetectEntitiesRequest = new BatchDetectEntitiesRequest()
                {
                    TextList = textToRetry,
                    LanguageCode = "en"
                };

                batchDetectEntitiesResponse =
comprehendClient.BatchDetectEntities(batchDetectEntitiesRequest);

                foreach(BatchDetectEntitiesItemResult item in
batchDetectEntitiesResponse.ResultList)
                {
                    Console.WriteLine("Entities in {0}:", textList[item.Index]);
                    foreach (Entity entity in item.Entities)
                        PrintEntity(entity);
                }
            }
            Console.WriteLine("End of DetectEntities");
```

```
        }
    }
}
```

Tâches d'analyse asynchrones à l'aide de l'API

Les exemples suivants utilisent le système asynchrone Amazon Comprehend APIs pour créer et gérer des tâches d'analyse à l'aide du. AWS CLI

Rubriques

- Analyse asynchrone pour Amazon Comprehend Insights
- Analyse asynchrone pour un sentiment ciblé
- Analyse asynchrone pour la détection d'événements
- Analyse asynchrone pour la modélisation de sujets

Analyse asynchrone pour Amazon Comprehend Insights

Les sections suivantes utilisent l'API Amazon Comprehend pour exécuter des opérations asynchrones afin d'analyser les informations d'Amazon Comprehend.

Rubriques

- Prérequis
- Démarrage d'une tâche d'analyse
- Tâches d'analyse de surveillance
- Obtenir des résultats d'analyse

Prérequis

Les documents doivent être dans des fichiers texte au format UTF-8. Vous pouvez soumettre vos documents dans deux formats. Le format que vous utilisez dépend du type de document que vous souhaitez analyser, comme décrit dans le tableau suivant.

Description	Format
Chaque fichier contient un document d'entrée. C'est la meilleure solution pour les collections de documents volumineux.	Un document par fichier
L'entrée est un ou plusieurs fichiers. Chaque ligne d'un fichier est considérée comme un document. C'est la meilleure solution pour les documents courts, tels que les publications sur les réseaux sociaux. Chaque ligne doit se terminer par une ligne d'alimentation (LF,\n), un retour en chariot (CR, \ r), ou les deux (CRLF, \ r\n). Vous ne pouvez pas utiliser le séparateur de ligne UTF-8 (u+2028) pour terminer une ligne.	Un document par ligne

Lorsque vous démarrez une tâche d'analyse, vous spécifiez l'emplacement S3 de vos données d'entrée. L'URI doit se trouver dans la même AWS région que le point de terminaison de l'API que vous appelez. L'URI peut pointer vers un seul fichier ou être le préfixe d'un ensemble de fichiers de données. Pour plus d'informations, consultez le type de données InputDataConfig.

Vous devez accorder à Amazon Comprehend l'accès au compartiment Amazon S3 qui contient votre collection de documents et vos fichiers de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Démarrage d'une tâche d'analyse

Pour soumettre une tâche d'analyse, utilisez la console Amazon Comprehend ou l'opération appropriée `Start*` :

- StartDominantLanguageDetectionJob— Lancez une tâche pour détecter la langue dominante dans chaque document de la collection. Pour plus d'informations sur la langue dominante d'un document, consultezLangue dominante.
- StartEntitiesDetectionJob— Lancez une tâche pour détecter les entités dans chaque document de la collection. Pour plus d'informations sur les entités, consultez Entités.

- StartKeyPhrasesDetectionJob— Lancez une tâche pour détecter les phrases clés dans chaque document de la collection. Pour plus d'informations sur les phrases clés, voirPhrases clés.
- StartPiiEntitiesDetectionJob— Commencez une tâche pour détecter les informations personnelles identifiables (PII) dans chaque document de la collection. Pour plus d'informations sur les informations personnelles, consultezDétection des entités PII.
- StartSentimentDetectionJob— Lancez une tâche pour détecter le sentiment qui se dégage de chaque document de la collection. Pour plus d'informations sur les sentiments, voirSentiment.

Tâches d'analyse de surveillance

L'`Start*`opération renvoie un identifiant que vous pouvez utiliser pour suivre la progression de la tâche.

Pour suivre la progression à l'aide de l'API, vous pouvez effectuer l'une des deux opérations suivantes, selon que vous souhaitez suivre la progression d'une tâche individuelle ou de plusieurs tâches.

Pour suivre la progression d'une tâche d'analyse individuelle, utilisez les `Describe*` opérations. Vous fournissez l'ID de tâche renvoyé par l'`Start*`opération. La réponse de l'`Describe*`opération contient le `JobStatus` champ indiquant le statut de la tâche.

Pour suivre la progression de plusieurs tâches d'analyse, utilisez les `List*` opérations. `List*`les opérations renvoient une liste des tâches que vous avez soumises à Amazon Comprehend. La réponse inclut un `JobStatus` champ pour chaque tâche qui indique le statut de la tâche.

Si le champ d'état est défini sur `COMPLETED` ou`FAILED`, le traitement de la tâche est terminé.

Pour obtenir le statut de tâches individuelles, utilisez l'`Describe*`opération correspondant à l'analyse que vous effectuez.

- DescribeDominantLanguageDetectionJob
- DescribeEntitiesDetectionJob
- DescribeKeyPhrasesDetectionJob
- DescribePiiEntitiesDetectionJob
- DescribeSentimentDetectionJob

Pour obtenir le statut de plusieurs tâches, utilisez l'`List*`opération correspondant à l'analyse que vous effectuez.

- ListDominantLanguageDetectionJobs
- ListEntitiesDetectionJobs
- ListKeyPhrasesDetectionJobs
- ListPiiEntitiesDetectionJobs
- ListSentimentDetectionJobs

Pour limiter les résultats aux tâches qui répondent à certains critères, utilisez le `Filter` paramètre `List*` des opérations. Vous pouvez filtrer en fonction du nom de la tâche, de son statut, ainsi que de la date et de l'heure de soumission de la tâche. Pour plus d'informations, consultez le `Filter` paramètre de chacune des `List*` opérations dans la référence de l'API Amazon Comprehend.

Obtenir des résultats d'analyse

Une fois le travail d'analyse terminé, utilisez une `Describe*` opération pour obtenir l'emplacement des résultats. Si le statut de la tâche est le suivant`COMPLETED`, la réponse inclut un `OutputDataConfig` champ contenant l'emplacement du fichier de sortie sur Amazon S3. Le fichier est une archive compressée qui contient les résultats de l'analyse. `output.tar.gz`

Si le statut d'une tâche est`FAILED`, la réponse contient un `Message` champ qui décrit la raison pour laquelle la tâche d'analyse ne s'est pas terminée correctement.

Pour obtenir le statut de chaque tâche, utilisez l'`Describe*`opération appropriée :

- DescribeDominantLanguageDetectionJob
- DescribeEntitiesDetectionJob
- DescribeKeyPhrasesDetectionJob
- DescribeSentimentDetectionJob

Les résultats sont renvoyés dans un seul fichier, avec une structure JSON pour chaque document. Chaque fichier de réponse inclut également des messages d'erreur pour toute tâche dont le champ d'état est défini sur`FAILED`.

Chacune des sections suivantes présente des exemples de sortie pour les deux formats d'entrée.

Obtenir des résultats de détection de la langue dominante

Voici un exemple de fichier de sortie issu d'une analyse qui a détecté la langue dominante. Le format de l'entrée est d'un document par ligne. Pour plus d'informations, consultez l'DetectDominantLanguageopération.

```
{"File": "0_doc", "Languages": [{"LanguageCode": "en", "Score": 0.9514502286911011},
 {"LanguageCode": "de", "Score": 0.02374090999364853}, {"LanguageCode": "nl", "Score":
 0.003208699868991971}, "Line": 0}
{"File": "1_doc", "Languages": [{"LanguageCode": "en", "Score": 0.9822712540626526},
 {"LanguageCode": "de", "Score": 0.002621392020955682}, {"LanguageCode": "es", "Score":
 0.002386554144322872}], "Line": 1}
```

Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier :

```
{"File": "small_doc", "Languages": [{"LanguageCode": "en", "Score":
 0.9728053212165833}, {"LanguageCode": "de", "Score": 0.007670710328966379},
 {"LanguageCode": "es", "Score": 0.002847236813977360 7}]}
{"File": "huge_doc", "Languages": [{"LanguageCode": "en", "Score": 0.984955906867981},
 {"LanguageCode": "de", "Score": 0.0026436643674969673}, {"LanguageCode": "fr",
 "Score": 0.0014206881169229746}]}
```

Obtenir les résultats de détection des entités

Voici un exemple de fichier de sortie issu d'une analyse qui a détecté des entités dans des documents. Le format de l'entrée est d'un document par ligne. Pour plus d'informations, consultez l'DetectEntitiesopération. La sortie contient deux messages d'erreur, l'un pour un document trop long et l'autre pour un document qui n'est pas au format UTF-8.

```
{"File": "50_docs", "Line": 0, "Entities": [{"BeginOffset": 0, "EndOffset": 22,
 "Score": 0.9763959646224976, "Text": "Cluj-NapocaCluj-Napoca", "Type": "LOCATION"}"]}
{"File": "50_docs", "Line": 1, "Entities": [{"BeginOffset": 11, "EndOffset": 15,
 "Score": 0.9615424871444702, "Text": "Maat", "Type": "PERSON"}}]}
{"File": "50_docs", "Line": 2, "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage":
 "Document size exceeds maximum size limit 102400 bytes."}
{"File": "50_docs", "Line": 3, "ErrorCode": "UNSUPPORTED_ENCODING", "ErrorMessage":
 "Document is not in UTF-8 format and all subsequent lines are ignored."}
```

Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier. La sortie contient deux messages d'erreur, l'un pour un document trop long et l'autre pour un document qui n'est pas au format UTF-8.


```
{"File": "non_utf8.txt", "ErrorCode": "UNSUPPORTED_ENCODING", "ErrorMessage": "Document
 is not in UTF-8 format and all subsequent line are ignored."}
{"File": "small_doc", "Entities": [{"BeginOffset": 0, "EndOffset": 4, "Score":
 0.645766019821167, "Text": "Maat", "Type": "PERSON"}]}
{"File": "huge_doc", "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage": "Document
 size exceeds size limit 102400 bytes."}
```


Obtenir les résultats de détection des phrases clés

Voici un exemple de fichier de sortie issu d'une analyse qui a détecté des phrases clés dans un document. Le format de l'entrée est d'un document par ligne. Pour plus d'informations, consultez l'DetectKeyPhrasesopération.


```
{"File": "50_docs", "KeyPhrases": [{"BeginOffset": 0, "EndOffset": 22, "Score":
 0.8948641419410706, "Text": "Cluj-NapocaCluj-Napoca"}, {"BeginOffset": 45,
 "EndOffset": 49, "Score": 0.9989854693412781, "Text": "Cluj"}], "Line": 0}
```


Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier.


```
{"File": "1_doc", "KeyPhrases": [{"BeginOffset": 0, "EndOffset": 22, "Score":
 0.8948641419410706, "Text": "Cluj-NapocaCluj-Napoca"}, {"BeginOffset": 45,
 "EndOffset": 49, "Score": 0.9989854693412781, "Text": "Cluj"}]}
```


Obtenir les résultats de détection des informations personnelles identifiables (PII)

Voici un exemple de fichier de sortie issu d'une tâche d'analyse qui a détecté des entités PII dans des documents. Le format de l'entrée est d'un document par ligne.


```
{"Entities":[{"Type":"NAME","BeginOffset":40,"EndOffset":69,"Score":0.999995},
{"Type":"ADDRESS","BeginOffset":247,"EndOffset":253,"Score":0.998828},
{"Type":"BANK_ACCOUNT_NUMBER","BeginOffset":406,"EndOffset":411,"Score":0.693283}],"File":"doc.
{"Entities":[{"Type":"SSN","BeginOffset":1114,"EndOffset":1124,"Score":0.999999},
{"Type":"EMAIL","BeginOffset":3742,"EndOffset":3775,"Score":0.999993},
{"Type":"PIN","BeginOffset":4098,"EndOffset":4102,"Score":0.999995}],"File":"doc.txt","Line":1}
```


Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier.

```
{"Entities":[{"Type":"NAME","BeginOffset":40,"EndOffset":69,"Score":0.999995},
{"Type":"ADDRESS","BeginOffset":247,"EndOffset":253,"Score":0.998828},
{"Type":"BANK_ROUTING","BeginOffset":279,"EndOffset":289,"Score":0.999999}],"File":"doc.txt"}
```


Obtenir des résultats de détection des sentiments

Voici un exemple de fichier de sortie issu d'une analyse qui a détecté le sentiment exprimé dans un document. Il inclut un message d'erreur car un document est trop long. Le format de l'entrée est d'un document par ligne. Pour plus d'informations, consultez l'DetectSentimentopération.


```
{"File": "50_docs", "Line": 0, "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed":
 0.002734508365392685, "Negative": 0.008935936726629734, "Neutral": 0.9841893315315247,
 "Positive": 0.004140198230743408}}
{"File": "50_docs", "Line": 1, "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage":
 "Document size is exceeded maximum size limit 5120 bytes."}
{"File": "50_docs", "Line": 2, "Sentiment": "NEUTRAL", "SentimentScore":
 {"Mixed": 0.0023119584657251835, "Negative": 0.0029857370536774397, "Neutral":
 0.9866572022438049, "Positive": 0.00804515462368726 7}}
```


Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier.


```
{"File": "small_doc", "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed":
 0.0023450672160834074, "Negative": 0.0009663937962614, "Neutral": 0.9795311689376831,
 "Positive": 0.01715377675175667}}
{"File": "huge_doc", "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage": "Document
 size is exceeds the limit of 5120 bytes."}
```


Analyse asynchrone pour un sentiment ciblé

Pour plus d'informations sur l'analyse en temps réel du sentiment ciblé, consultezthe section called “Analyse en temps réel pour un sentiment ciblé”.

Amazon Comprehend fournit les opérations d'API suivantes pour démarrer et gérer une analyse ciblée asynchrone des sentiments :

- StartTargetedSentimentDetectionJob— Démarre une tâche de détection ciblée asynchrone des sentiments pour un ensemble de documents.
- ListTargetedSentimentDetectionJobs— Renvoie la liste des tâches de détection de sentiments ciblées que vous avez soumises.

- DescribeTargetedSentimentDetectionJob— Obtient les propriétés (y compris le statut) associées à la tâche de détection de sentiments ciblée spécifiée.
- StopTargetedSentimentDetectionJob— Arrête la tâche de sentiment ciblée en cours spécifiée.

Rubriques

- Avant de commencer
- Analyser le sentiment ciblé à l'aide du AWS CLI

Avant de commencer

Avant de commencer, assurez-vous que vous disposez des éléments suivants :

- Compartiments d'entrée et de sortie : identifiez les compartiments Amazon S3 que vous souhaitez utiliser pour les entrées et les sorties. Les buckets doivent se trouver dans la même région que l'API que vous appelez.
- Rôle de service IAM : vous devez disposer d'un rôle de service IAM autorisé à accéder à vos compartiments d'entrée et de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Analyser le sentiment ciblé à l'aide du AWS CLI

L'exemple suivant montre comment utiliser l'`StartTargetedSentimentDetectionJob`opération avec le AWS CLI. Cet exemple indique la langue du texte saisi.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend start-targeted-sentiment-detection-job \
       --job-name "job name" \
       --language-code "en" \
       --cli-input-json file://path to JSON input file
```

Pour le `cli-input-json` paramètre, vous fournissez le chemin d'accès à un fichier JSON contenant les données de demande, comme illustré dans l'exemple suivant.

```
{
```

```
    "InputDataConfig": {
        "S3Uri": "s3://input bucket/input path",
        "InputFormat": "ONE_DOC_PER_FILE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://output bucket/output path"
    },
    "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"
}
```

Si la demande de démarrage de la tâche a été acceptée, vous recevrez la réponse suivante :

```
{
    "JobStatus": "SUBMITTED",
    "JobArn": "job ARN"
    "JobId": "job ID"
}
```

Analyse asynchrone pour la détection d'événements

Rubriques

- Avant de commencer
- Détectez les événements à l'aide du AWS CLI
- Répertoriez les événements en utilisant le AWS CLI
- Décrivez les événements à l'aide du AWS CLI
- Obtenez les résultats de détection des événements

Pour détecter des événements dans un ensemble de documents, utilisez le StartEventsDetectionJobpour démarrer une tâche asynchrone.

Avant de commencer

Avant de commencer, assurez-vous que vous disposez des éléments suivants :

- Compartiments d'entrée et de sortie : identifiez les compartiments Amazon S3 que vous souhaitez utiliser pour les entrées et les sorties. Les buckets doivent se trouver dans la même région que l'API que vous appelez.

- Rôle de service IAM : vous devez disposer d'un rôle de service IAM autorisé à accéder à vos compartiments d'entrée et de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Détectez les événements à l'aide du AWS CLI

L'exemple suivant illustre l'utilisation de l'StartEventsDetectionJobopération avec AWS CLI

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend start-events-detection-job \
  --region region \
  --job-name job name \
  --cli-input-json file://path to JSON input file
```

Pour le `cli-input-json` paramètre, vous fournissez le chemin d'accès à un fichier JSON contenant les données de demande, comme illustré dans l'exemple suivant.

```
{
  "InputDataConfig": {
      "S3Uri": "s3://input bucket/input path",
      "InputFormat": "ONE_DOC_PER_LINE"
  },
  "OutputDataConfig": {
      "S3Uri": "s3://output bucket/output path"
  },
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"
  "LanguageCode": "en",
  "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "INVESTMENT_GENERAL",
      "CORPORATE_MERGER",
      "IPO",
      "RIGHTS_ISSUE",
      "SECONDARY_OFFERING",
      "SHELF_OFFERING",
      "TENDER_OFFERING",
```

```
      "STOCK_SPLIT"
  ]
}
```

Si la demande de démarrage de la tâche de détection des événements est réussie, vous recevrez la réponse suivante :

```
{
  "JobStatus": "SUBMITTED",
  "JobId": "job ID"
}
```

Répertoriez les événements en utilisant le AWS CLI

Utilisez cette ListEventsDetectionJobsopération pour voir la liste des tâches de détection d'événements que vous avez soumises. La liste inclut des informations sur les emplacements d'entrée et de sortie que vous avez utilisés et sur l'état de chacune des tâches de détection. L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend list-events-detection-jobs --region region
```

Vous obtiendrez un JSON similaire à ce qui suit en réponse :

```
{
 "EventsDetectionJobPropertiesList": [
    {
       "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role",
       "EndTime": timestamp,
       "InputDataConfig": {
          "InputFormat": "ONE_DOC_PER_LINE",
          "S3Uri": "s3://input bucket/input path"
       },
       "JobId": "job ID",
       "JobName": "job name",
       "JobStatus": "COMPLETED",
       "LanguageCode": "en",
       "Message": "message",
       "OutputDataConfig": {
```

```
          "S3Uri": "s3://output bucket/ouput path"
       },
       "SubmitTime": timestamp,
       "TargetEventTypes": [
         "BANKRUPTCY",
         "EMPLOYMENT",
         "CORPORATE_ACQUISITION",
         "INVESTMENT_GENERAL",
         "CORPORATE_MERGER",
         "IPO",
         "RIGHTS_ISSUE",
         "SECONDARY_OFFERING",
         "SHELF_OFFERING",
         "TENDER_OFFERING",
         "STOCK_SPLIT"
  ]
    }
 ],
 "NextToken": "next token"
}
```

Décrivez les événements à l'aide du AWS CLI

Vous pouvez utiliser cette DescribeEventsDetectionJobopération pour obtenir le statut d'une tâche existante. L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend describe-events-detection-job \
  --region region \
  --job-id job ID
```

Vous obtiendrez le JSON suivant en réponse :

```
{
 "EventsDetectionJobProperties": {
    "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role",
    "EndTime": timestamp,
    "InputDataConfig": {
       "InputFormat": "ONE_DOC_PER_LINE",
       "S3Uri": "S3Uri": "s3://input bucket/input path"
```

```
    },
    "JobId": "job ID",
    "JobName": "job name",
    "JobStatus": "job status",
    "LanguageCode": "en",
    "Message": "message",
    "OutputDataConfig": {
       "S3Uri": "s3://output bucket/output path"
    },
    "SubmitTime": timestamp,
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "INVESTMENT_GENERAL",
      "CORPORATE_MERGER",
      "IPO",
      "RIGHTS_ISSUE",
      "SECONDARY_OFFERING",
      "SHELF_OFFERING",
      "TENDER_OFFERING",
      "STOCK_SPLIT"
  ]
 }
}
```

Obtenez les résultats de détection des événements

Voici un exemple de fichier de sortie issu d'une tâche d'analyse qui a détecté des événements dans des documents. Le format de l'entrée est d'un document par ligne.

```
{"Entities": [{"Mentions": [{"BeginOffset": 12, "EndOffset": 27, "GroupScore":
 1.0, "Score": 0.916355, "Text": "over a year ago", "Type": "DATE"}]}, {"Mentions":
 [{"BeginOffset": 33, "EndOffset": 39, "GroupScore": 1.0, "Score": 0.996603,
 "Text": "Amazon", "Type": "ORGANIZATION"}]}, {"Mentions": [{"BeginOffset":
 66, "EndOffset": 77, "GroupScore": 1.0, "Score": 0.999283, "Text": "Whole
 Foods", "Type": "ORGANIZATION"}]}], "Events": [{"Arguments": [{"EntityIndex":
 2, "Role": "INVESTEE", "Score": 0.999283}, {"EntityIndex": 0, "Role": "DATE",
 "Score": 0.916355}, {"EntityIndex": 1, "Role": "INVESTOR", "Score": 0.996603}],
 "Triggers": [{"BeginOffset": 373, "EndOffset": 380, "GroupScore": 0.999984,
 "Score": 0.999955, "Text": "acquire", "Type": "CORPORATE_ACQUISITION"}], "Type":
 "CORPORATE_ACQUISITION"}, {"Arguments": [{"EntityIndex": 2, "Role": "PARTICIPANT",
 "Score": 0.999283}], "Triggers": [{"BeginOffset": 115, "EndOffset": 123, "GroupScore":
```

```
1.0, "Score": 0.999967, "Text": "combined", "Type": "CORPORATE_MERGER"}], "Type":
"CORPORATE_MERGER"}], "File": "doc.txt", "Line": 0}
```

Pour plus d'informations sur la structure du fichier de sortie des événements et les types d'événements pris en charge, consultezÉvénements.

Analyse asynchrone pour la modélisation de sujets

Pour déterminer les sujets d'un ensemble de documents, utilisez le StartTopicsDetectionJobpour démarrer une tâche asynchrone. Vous pouvez suivre des sujets dans des documents rédigés en anglais ou en espagnol.

Rubriques

- Avant de commencer
- À l'aide du AWS Command Line Interface
- À l'aide du SDK pour Python ou SDK pour .NET

Avant de commencer

Avant de commencer, assurez-vous que vous disposez des éléments suivants :

- Compartiments d'entrée et de sortie : identifiez les compartiments Amazon S3 que vous souhaitez utiliser pour les entrées et les sorties. Les buckets doivent se trouver dans la même région que l'API que vous appelez.
- Rôle de service IAM : vous devez disposer d'un rôle de service IAM autorisé à accéder à vos compartiments d'entrée et de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

À l'aide du AWS Command Line Interface

L'exemple suivant illustre l'utilisation de l'`StartTopicsDetectionJob`opération avec AWS CLI

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend start-topics-detection-job \
```

```
    --number-of-topics topics to return \
    --job-name "job name" \
    --region region \
    --cli-input-json file://path to JSON input file
```

Pour le `cli-input-json` paramètre, vous fournissez le chemin d'accès à un fichier JSON contenant les données de demande, comme illustré dans l'exemple suivant.

```
{
    "InputDataConfig": {
        "S3Uri": "s3://input bucket/input path",
        "InputFormat": "ONE_DOC_PER_FILE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://output bucket/output path"
    },
    "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"
}
```

Si la demande de démarrage de la tâche de détection de sujets aboutit, vous recevrez la réponse suivante :

```
{
    "JobStatus": "SUBMITTED",
    "JobId": "job ID"
}
```

Utilisez cette ListTopicsDetectionJobsopération pour afficher la liste des tâches de détection de sujets que vous avez soumises. La liste inclut des informations sur les emplacements d'entrée et de sortie que vous avez utilisés et sur l'état de chacune des tâches de détection. L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend list-topics-detection-jobs \-- region
```

Vous obtiendrez un JSON similaire à ce qui suit en réponse :

```
{
    "TopicsDetectionJobPropertiesList": [
        {
```

```
            "InputDataConfig": {
                "S3Uri": "s3://input bucket/input path",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "NumberOfTopics": topics to return,
            "JobId": "job ID",
            "JobStatus": "COMPLETED",
            "JobName": "job name",
            "SubmitTime": timestamp,
            "OutputDataConfig": {
                "S3Uri": "s3://output bucket/output path"
            },
            "EndTime": timestamp
        },
        {
            "InputDataConfig": {
                "S3Uri": "s3://input bucket/input path",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "NumberOfTopics": topics to return,
            "JobId": "job ID",
            "JobStatus": "RUNNING",
            "JobName": "job name",
            "SubmitTime": timestamp,
            "OutputDataConfig": {
                "S3Uri": "s3://output bucket/output path"
            }
        }
    ]
}
```

Vous pouvez utiliser cette DescribeTopicsDetectionJobopération pour obtenir le statut d'une tâche existante. L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend describe-topics-detection-job --job-id job ID
```

Vous obtiendrez le JSON suivant en réponse :

```
{
    "TopicsDetectionJobProperties": {
```

```
        "InputDataConfig": {
            "S3Uri": "s3://input bucket/input path",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "NumberOfTopics": topics to return,
        "JobId": "job ID",
        "JobStatus": "COMPLETED",
        "JobName": "job name",
        "SubmitTime": timestamp,
        "OutputDataConfig": {
            "S3Uri": "s3://output bucket/ouput path"
        },
        "EndTime": timestamp
    }
}
```

À l'aide du SDK pour Python ou SDK pour .NET

Pour obtenir des exemples de SDK expliquant comment démarrer une tâche de modélisation de rubriques, consultezUtilisation `StartTopicsDetectionJob` avec un AWS SDK ou une CLI.

Confiance et sécurité

Les utilisateurs génèrent de grandes quantités de contenu textuel par le biais d'applications en ligne (telles que peer-to-peer les chats et les forums de discussion), de commentaires publiés sur des sites Web et d'applications d'IA génératives (instructions de saisie et sorties issues de modèles d'IA génératifs). Les fonctionnalités de confiance et de sécurité d'Amazon Comprehend peuvent vous aider à modérer ce contenu afin de fournir un environnement sûr et inclusif à vos utilisateurs.

Les avantages de l'utilisation des fonctionnalités de confiance et de sécurité d'Amazon Comprehend incluent :

- Modération plus rapide : modérez rapidement et précisément un grand volume de texte pour protéger vos plateformes en ligne de tout contenu inapproprié.
- Personnalisable : personnalisez les seuils de modération dans les réponses des API en fonction des besoins de votre application.
- Facile à utiliser : configurez les fonctionnalités de confiance et de sécurité par le biais de LangChain l'intégration ou à l'aide du AWS CLI ou SDKs.

La confiance et la sécurité d'Amazon Comprehend abordent les aspects suivants de la modération du contenu :

- Toxicity detection— Détectez les contenus susceptibles d'être dangereux, offensants ou inappropriés. Les exemples incluent les discours de haine, les menaces ou les abus.
- Intent classification— Détectez le contenu ayant une intention malveillante explicite ou implicite. Les exemples incluent le contenu discriminatoire ou illégal, ou le contenu qui exprime ou demande des conseils sur des sujets médicaux, juridiques, politiques, controversés, personnels ou financiers.
- Privacy protection— Les utilisateurs peuvent fournir par inadvertance du contenu susceptible de révéler des informations personnelles identifiables (PII). Amazon Comprehend PII permet de détecter et de supprimer des informations personnelles.

Rubriques

- Détection de toxicité
- Classification de sécurité rapide
- Détection et rédaction des informations personnelles

Détection de toxicité

La détection de toxicité d'Amazon Comprehend permet de détecter en temps réel le contenu toxique lors d'interactions basées sur du texte. Vous pouvez utiliser la détection de toxicité pour modérer peer-to-peer les conversations sur les plateformes en ligne ou pour surveiller les entrées et sorties génératives de l'IA.

La détection de toxicité permet de détecter les catégories de contenu offensant suivantes :

GRAPHIC

Le discours graphique utilise des images visuellement descriptives, détaillées et d'une vivacité désagréable. Un tel langage est souvent verbeux pour amplifier une insulte, un malaise ou un préjudice pour le destinataire.

HARCÈLEMENT_OU_ABUS

Un discours qui impose une dynamique de pouvoir perturbatrice entre le locuteur et l'auditeur, quelle que soit l'intention, cherche à affecter le bien-être psychologique du destinataire ou à objectiver une personne.

DISCOURS DE HAINE

Discours qui critique, insulte, dénonce ou déshumanise une personne ou un groupe sur la base d'une identité, qu'il s'agisse de sa race, de son origine ethnique, de son identité de genre, de sa religion, de son orientation sexuelle, de ses capacités, de son origine nationale ou d'un autre groupe identitaire.

INSULTE

Discours comportant un langage dégradant, humiliant, moqueur, insultant ou dénigrant.

BLASPHÈME

Un discours contenant des mots, des phrases ou des acronymes impolis, vulgaires ou offensants est considéré comme profane.

SEXUELLE

Discours qui indique un intérêt, une activité ou une excitation sexuels en utilisant des références directes ou indirectes à des parties du corps, à des traits physiques ou au sexe.

VIOLENCE_OU_MENACE

Discours comportant des menaces visant à infliger de la douleur, des blessures ou de l'hostilité à une personne ou à un groupe.

TOXICITÉ

Discours contenant des mots, des phrases ou des acronymes susceptibles d'être considérés comme toxiques par nature dans l'une des catégories ci-dessus.

Détection du contenu toxique à l'aide de l'API

Pour détecter un contenu toxique dans un texte, utilisez l'DetectToxicContentopération synchrone. Cette opération effectue une analyse sur une liste de chaînes de texte que vous fournissez en entrée. La réponse de l'API contient une liste de résultats dont la taille correspond à celle de la liste d'entrée.

Actuellement, la détection de contenu toxique ne prend en charge que la langue anglaise. Pour le texte saisi, vous pouvez fournir une liste de 10 chaînes de texte au maximum. La taille maximale de chaque chaîne est de 1 Ko.

La détection du contenu toxique renvoie une liste des résultats d'analyse, une entrée dans la liste pour chaque chaîne d'entrée. Une entrée contient une liste des types de contenu toxique identifiés dans la chaîne de texte, ainsi qu'un score de confiance pour chaque type de contenu. L'entrée inclut également un score de toxicité pour la chaîne.

Les exemples suivants montrent comment utiliser l'`DetectToxicContent`opération à l'aide de Python AWS CLI et.

AWS CLI

Vous pouvez détecter un contenu toxique à l'aide de la commande suivante dans AWS CLI :

```
aws comprehend detect-toxic-content --language-code en  /
            --text-segments "[{\"Text\":\"You are so obtuse\"}]"
```

Il AWS CLI répond avec le résultat suivant. Le segment de texte reçoit un score de confiance élevé dans la `INSULT` catégorie, avec pour résultat un score de toxicité élevé :

```
{
   "ResultList": [
      {
```

```
         "Labels": [
                {
                    "Name": "PROFANITY",
                    "Score": 0.0006000000284984708
                },
                {
                    "Name": "HATE_SPEECH",
                    "Score": 0.00930000003427267
                },
                {
                    "Name": "INSULT",
                    "Score": 0.9204999804496765
                },
                {
                    "Name": "GRAPHIC",
                    "Score": 9.999999747378752e-05
                },
                {
                    "Name": "HARASSMENT_OR_ABUSE",
                    "Score": 0.0052999998442828655
                },
                {
                    "Name": "SEXUAL",
                    "Score": 0.01549999974668026
                },
                {
                    "Name": "VIOLENCE_OR_THREAT",
                    "Score": 0.007799999788403511
                }
            ],
            "Toxicity": 0.7192999720573425
      }
   ]
}
```

Vous pouvez saisir jusqu'à 10 chaînes de texte, en utilisant le format suivant pour le `text-segments` paramètre :

```
   --text-segments "[{\"Text\":\"text string 1\"},
                    {\"Text\":\"text string2\"},
                    {\"Text\":\"text string3\"}]"
```

La AWS CLI réponse donne les résultats suivants :

```
{
   "ResultList": [
      {
         "Labels": [ (truncated) ],
            "Toxicity": 0.3192999720573425
      },
      {
         "Labels": [ (truncated) ],
            "Toxicity": 0.1192999720573425
      },
      {
         "Labels": [ (truncated) ],
            "Toxicity": 0.0192999720573425
      }
   ]
}
```

Python (Boto)

L'exemple suivant montre comment détecter un contenu toxique à l'aide de Python :

```
import boto3
client = boto3.client(
    service_name='comprehend',
    region_name=region) # For example, 'us-west-2'

response = client.detect_toxic_content(
    LanguageCode='en',
    TextSegments=[{'Text': 'You are so obtuse'}]
)
print("Response: %s\n" % response)
```

Classification de sécurité rapide

Amazon Comprehend fournit un classificateur binaire préentraîné permettant de classer les demandes de saisie en texte brut pour les grands modèles linguistiques (LLM) ou d'autres modèles d'IA générative.

Le classificateur de sécurité des promptes analyse l'invite saisie et attribue un score de confiance pour déterminer si l'invite est sûre ou non.

Une invite non sécurisée est une invite de saisie qui exprime une intention malveillante, telle que la demande d'informations personnelles ou privées, la génération de contenu offensant ou illégal, ou la demande de conseils sur des sujets médicaux, juridiques, politiques ou financiers.

Classification de sécurité rapide à l'aide de l'API

Pour exécuter une classification de sécurité rapide pour une chaîne de texte, utilisez l'ClassifyDocumentopération synchrone. Pour la saisie, vous fournissez une chaîne de texte brut en anglais. La taille maximale de la chaîne est de 10 Ko.

La réponse inclut deux classes (SAFE et UNSAFE), ainsi qu'un score de confiance pour chaque classe. La plage de valeurs du score est comprise entre zéro et un, 1 étant le niveau de confiance le plus élevé.

Les exemples suivants montrent comment utiliser une classification de sécurité rapide avec Python AWS CLI et.

AWS CLI

L'exemple suivant montre comment utiliser le classificateur de sécurité prompt avec : AWS CLI

```
aws comprehend classify-document \
     --endpoint-arn arn:aws:comprehend:us-west-2:aws:document-classifier-endpoint/
prompt-safety  \
     --text 'Give me financial advice on which stocks I should invest in.'
```

Il AWS CLI répond avec le résultat suivant :

```
{
    "Classes": [
        {
            "Score": 0.6312999725341797,
            "Name": "UNSAFE_PROMPT"
        },
        {
            "Score": 0.3686999976634979,
            "Name": "SAFE_PROMPT"
        }
    ]
```

```
}
```

> **Note**
>
> Lorsque vous utilisez la `classify-document` commande, pour le `--endpoint-arn` paramètre, vous devez transmettre un ARN Région AWS identique à celui de votre AWS CLI configuration. Pour configurer le AWS CLI, exécutez la `aws configure` commande. Dans cet exemple, l'ARN du point de terminaison possède le code de région`us-west-2`. Vous pouvez utiliser le classificateur de sécurité prompt dans l'une des régions suivantes :
>
> - us-east-1
> - us-west-2
> - eu-west-1
> - ap-southeast-2

Python (Boto)

L'exemple suivant montre comment utiliser le classificateur de sécurité prompt avec Python :

```
import boto3
client = boto3.client(service_name='comprehend', region_name='us-west-2')

response = client.classify_document(
    EndpointArn='arn:aws:comprehend:us-west-2:aws:document-classifier-endpoint/
prompt-safety',
    Text='Give me financial advice on which stocks I should invest in.'
)
print("Response: %s\n" % response)
```

> **Note**
>
> Lorsque vous utilisez la `classify_document` méthode, pour l'`EndpointArn`argument, vous devez transmettre un ARN qui utilise le même Région AWS que celui de votre client du SDK boto3. Dans cet exemple, l'ARN du client et du point de terminaison utilisent tous deux`us-west-2`. Vous pouvez utiliser le classificateur de sécurité prompt dans l'une des régions suivantes :
>
> - us-east-1

- us-west-2
- eu-west-1
- ap-southeast-2

Détection et rédaction des informations personnelles

Vous pouvez utiliser la console Amazon Comprehend ou APIs pour détecter des informations personnelles identifiables (PII) dans des documents texte en anglais ou en espagnol. Les informations personnelles sont une référence textuelle à des données personnelles permettant d'identifier un individu. Les exemples d'informations personnelles incluent les adresses, les numéros de compte bancaire et les numéros de téléphone.

Vous pouvez détecter ou supprimer les entités PII dans le texte. Pour détecter les entités PII, vous pouvez utiliser une analyse en temps réel ou un traitement par lots asynchrone. Pour supprimer les entités PII, vous devez utiliser un traitement par lots asynchrone.

Pour de plus amples informations, veuillez consulter données d'identification personnelle (PII) .

données d'identification personnelle (PII)

Vous pouvez utiliser la console Amazon Comprehend ou APIs pour détecter des informations personnelles identifiables (PII) dans des documents texte en anglais ou en espagnol. Les informations personnelles sont une référence textuelle à des données personnelles qui pourraient être utilisées pour identifier un individu. Les exemples d'informations personnelles incluent les adresses, les numéros de compte bancaire et les numéros de téléphone.

Avec la détection des informations personnelles, vous avez le choix de localiser les entités d'informations personnelles ou de les supprimer dans le texte. Pour localiser les entités PII, vous pouvez utiliser une analyse en temps réel ou un traitement par lots asynchrone. Pour supprimer les entités PII, vous devez utiliser un traitement par lots asynchrone.

Vous pouvez utiliser les points d'accès Amazon S3 Object Lambda pour les informations personnelles identifiables (PII) afin de contrôler la récupération de documents depuis votre compartiment Amazon S3. Vous pouvez contrôler l'accès aux documents contenant des informations personnelles et supprimer les informations personnelles identifiables des documents. Pour de plus amples informations, veuillez consulter Utilisation des points d'accès Lambda des objets Amazon S3 pour les informations personnelles identifiables (PII).

Rubriques

- Détection des entités PII
- Étiquetage des entités PII
- Analyse en temps réel des informations personnelles (console)
- Tâches d'analyse asynchrones PII (console)
- Analyse en temps réel des informations personnelles (API)
- Tâches d'analyse asynchrones PII (API)

Détection des entités PII

Vous pouvez utiliser Amazon Comprehend pour détecter les entités PII dans des documents texte en anglais ou en espagnol. Une entité PII est un type spécifique d'informations personnelles identifiables (PII). Utilisez la détection des PII pour localiser les entités PII ou supprimer les entités PII dans le texte.

Rubriques

- Localiser les entités PII
- Rédiger les entités PII
- Types d'entités universelles PII
- Types d'entités PII spécifiques à chaque pays

Localiser les entités PII

Pour localiser les entités PII dans votre texte, vous pouvez analyser rapidement un seul document à l'aide d'une analyse en temps réel. Vous pouvez également démarrer un traitement par lots asynchrone sur un ensemble de documents.

Vous pouvez utiliser la console ou l'API pour analyser en temps réel un seul document. Le texte saisi peut inclure jusqu'à 100 kilo-octets de caractères codés en UTF-8.

Par exemple, vous pouvez envoyer le texte d'entrée suivant pour localiser les entités PII :

Bonjour Paulo Santos. Le dernier relevé de votre compte de carte de crédit 1111-0000-1111-0000 a été envoyé par la poste au 123 Any Street, Seattle, WA 98109.

La sortie inclut les informations selon lesquelles « Paul Santos » a le type`NAME`, « 1111-0000-1111-0000 » a le type et « 123 Any Street`CREDIT_DEBIT_NUMBER`, Seattle, WA 98109 » a le type. `ADDRESS`

Amazon Comprehend renvoie une liste des entités PII détectées, avec les informations suivantes pour chaque entité PII :

- Un score qui estime la probabilité que l'étendue de texte détectée soit le type d'entité détecté.
- Type d'entité PII.
- L'emplacement de l'entité PII dans le document, spécifié sous forme de décalages de caractères pour le début et la fin de l'entité.

Par exemple, le texte d'entrée mentionné précédemment produit la réponse suivante :

```
{
    "Entities": [
        {
```

```
            "Score": 0.9999669790267944,
            "Type": "NAME",
            "BeginOffset": 6,
            "EndOffset": 18
        },
        {
            "Score": 0.8905550241470337,
            "Type": "CREDIT_DEBIT_NUMBER",
            "BeginOffset": 69,
            "EndOffset": 88
        },
        {
            "Score": 0.9999889731407166,
            "Type": "ADDRESS",
            "BeginOffset": 103,
            "EndOffset": 138
        }
    ]
}
```

Rédiger les entités PII

Pour supprimer les entités PII de votre texte, vous pouvez utiliser la console ou l'API pour démarrer un traitement par lots asynchrone. Amazon Comprehend renvoie une copie du texte saisi avec des modifications rédactionnelles pour chaque entité PII.

Par exemple, vous pouvez envoyer le texte d'entrée suivant pour supprimer les entités PII :

Bonjour Paulo Santos. Le dernier relevé de votre compte de carte de crédit 1111-0000-1111-0000 a été envoyé par la poste au 123 Any Street, Seattle, WA 98109.

Le fichier de sortie contient le texte suivant :

Bonjour ***** ******. Le dernier relevé de votre compte de carte de crédit ********** a été envoyé à *** *** ******* ******** ** *****.

Types d'entités universelles PII

Certains types d'entités PII sont universels (ils ne sont pas spécifiques à chaque pays), tels que les adresses e-mail et les numéros de carte de crédit. Amazon Comprehend détecte les types d'entités PII universelles suivants :

ADDRESS

Une adresse physique, telle que « 100 Main Street, Anytown, États-Unis » ou « Suite #12, Building 123 ». Une adresse peut inclure des informations telles que la rue, le bâtiment, l'emplacement, la ville, l'État, le pays, le comté, le code postal, le quartier et le quartier.

AGE

L'âge d'une personne, y compris la quantité et l'unité de temps. Par exemple, dans la phrase « J'ai 40 ans », Amazon Comprehend reconnaît « 40 ans » comme un âge.

AWS_ACCESS_CLÉ

Identifiant unique associé à une clé d'accès secrète ; vous utilisez l'ID de clé d'accès et la clé d'accès secrète pour signer des AWS demandes programmatiques de manière cryptographique.

AWS_SECRET_CLÉ

Identifiant unique associé à une clé d'accès. Vous utilisez l'ID de clé d'accès et la clé d'accès secrète pour signer les AWS demandes programmatiques de manière cryptographique.

CREDIT_DEBIT_CVV

Code de vérification de carte (CVV) à trois chiffres présent sur les cartes de crédit et de débit VISA et Discover. MasterCard Pour les cartes de crédit ou de débit American Express, le CVV est un code numérique à quatre chiffres.

EXPIRATION_DEBIT_CRÉDIT/EXPIRATION

Date d'expiration d'une carte de crédit ou de débit. Ce numéro est généralement composé de quatre chiffres et est souvent formaté au format month/year MM/YY. Amazon Comprehend reconnaît les dates d'expiration telles que le 21/01, le 01/2021 et le 1er janvier 2021.

NUMÉRO DE CRÉDIT_DÉBIT

Numéro de carte de crédit ou de débit. La longueur de ces numéros peut varier de 13 à 16 chiffres. Toutefois, Amazon Comprehend reconnaît également les numéros de carte de crédit ou de débit lorsque seuls les quatre derniers chiffres sont présents.

DATE_HEURE

Une date peut inclure une année, un mois, un jour, un jour de la semaine ou une heure de la journée. Par exemple, Amazon Comprehend reconnaît les dates « 19 janvier 2020 » ou « 11 h ». Amazon Comprehend reconnaîtra les dates partielles, les plages de dates et les intervalles de dates. Il reconnaîtra également des décennies, telles que « les années 1990 ».

IDENTIFIANT_PILOTE

Le numéro attribué à un permis de conduire, qui est un document officiel permettant à une personne de conduire un ou plusieurs véhicules motorisés sur la voie publique. Le numéro de permis de conduire est composé de caractères alphanumériques.

EMAIL

Une adresse e-mail, telle que marymajor@email.com.

NUMÉRO_COMPTE_BANCAIRE INTERNATIONAL

Un numéro de compte bancaire international a des formats spécifiques dans chaque pays. Consultez le site www.iban.com/structure.

IP_ADDRESS

Une IPv4 adresse, telle que 198.51.100.0.

PLAQUE D'IMMATRICULATION

La plaque d'immatriculation d'un véhicule est délivrée par l'État ou le pays où le véhicule est immatriculé. Le format des véhicules de tourisme est généralement composé de cinq à huit chiffres, composés de lettres majuscules et de chiffres. Le format varie en fonction de la localisation de l'État ou du pays émetteur.

MAC_ADDRESS

Une adresse MAC (Media Access Control) est un identifiant unique attribué à un contrôleur d'interface réseau (NIC).

NAME

Nom d’une personne. Ce type d'entité n'inclut pas les titres tels que Dr., Mr., Mrs., ou Miss. Amazon Comprehend n'applique pas ce type d'entité aux noms qui font partie d'organisations ou d'adresses. Par exemple, Amazon Comprehend reconnaît « John Doe Organization » comme une organisation, et reconnaît « Jane Doe Street » comme une adresse.

PASSWORD

Chaîne alphanumérique utilisée comme mot de passe, telle que « *very20special #pass * ».

PHONE

Numéro de téléphone. Ce type d’entité inclut également les numéros de télécopieur et de pager.

BROCHE

Numéro d'identification personnel (PIN) à quatre chiffres avec lequel vous pouvez accéder à votre compte bancaire.

CODE SWIFT_CODE

Un code SWIFT est un format standard de code d'identification bancaire (BIC) utilisé pour spécifier une banque ou une succursale en particulier. Les banques utilisent ces codes pour les transferts d'argent tels que les virements internationaux.

Les codes SWIFT sont composés de 8 ou 11 caractères. Les codes à 11 chiffres font référence à des succursales spécifiques, tandis que les codes à huit chiffres (ou les codes à 11 chiffres se terminant par « XXX ») font référence au siège social ou au bureau principal.

URL

Une adresse Web, telle que www.example.com.

USERNAME

Nom d'utilisateur identifiant un compte, tel qu'un nom de connexion, un pseudonyme, un surnom ou un pseudo.

NUMÉRO_D'IDENTIFICATION_VÉHICULE

Un numéro d'identification du véhicule (VIN) identifie un véhicule de manière unique. Le contenu et le format du VIN sont définis dans la spécification ISO 3779. Chaque pays possède des codes et des formats spécifiques pour VINs.

Types d'entités PII spécifiques à chaque pays

Certains types d'entités PII sont spécifiques à un pays, tels que les numéros de passeport et autres numéros d'identification émis par le gouvernement. Amazon Comprehend détecte les types suivants d'entités PII spécifiques à un pays :

CA_HEALTH_NUMBER

Un numéro de service de santé canadien est un identifiant unique à 10 chiffres, requis pour que les personnes aient accès aux prestations de santé.

CA_SOCIAL_INSURANCE_NUMBER

Le numéro d'assurance sociale (SIN) canadien est un identifiant unique à neuf chiffres, requis pour que les personnes aient accès aux programmes et aux prestations du gouvernement.

Le SIN est formaté en trois groupes de trois chiffres, tels que 123-456-789. Un SIN peut être validé au moyen d'un simple processus à chiffres de contrôle appelé algorithme de Luhn.

IN_AADHAAR

Un Aadhaar indien est un numéro d'identification unique à 12 chiffres délivré par le gouvernement indien aux résidents de l'Inde. Le format Aadhaar comporte un espace ou un trait d'union après le quatrième et le huitième chiffre.

IN_NREGA

Un numéro de la Loi nationale indienne sur la garantie de l'emploi rural (NREGA) se compose de deux lettres suivies de 14 chiffres.

IN_PERMANENT_ACCOUNT_NUMBER

Un numéro de compte permanent indien est un numéro alphanumérique unique à 10 chiffres émis par le service de l'impôt sur le revenu.

IN_VOTER_NUMBER

Un identifiant d'électeur indien se compose de trois lettres suivies de sept chiffres.

UK_NATIONAL_HEALTH_SERVICE_NUMBER

Un numéro du National Health Service du Royaume-Uni est un numéro de 10 à 17 chiffres, tel que le 485 777 3456. Le système actuel met en forme le numéro à 10 chiffres avec des espaces après les troisième et sixième chiffres. Le dernier chiffre est une somme de contrôle détectant les erreurs.

Le format numérique à 17 chiffres comporte des espaces après les 10e et 13e chiffres.

UK_NATIONAL_INSURANCE_NUMBER

Un numéro d'assurance nationale britannique (NINO) permet aux particuliers d'accéder aux prestations de l'assurance nationale (sécurité sociale). Il est également utilisé à certaines fins dans le système fiscal britannique.

Le numéro est composé de neuf chiffres et commence par deux lettres, suivies de six chiffres et d'une lettre. Un NINO peut être formaté avec un espace ou un tiret après les deux lettres et après les deuxième, quatrième et sixième chiffres.

UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER

Une référence unique de contribuable britannique (UTR) est un numéro à 10 chiffres qui identifie un contribuable ou une entreprise.

NUMÉRO_COMPTE_BANCAIRE

Un numéro de compte bancaire américain, généralement composé de 10 à 12 chiffres. Amazon Comprehend reconnaît également les numéros de compte bancaire lorsque seuls les quatre derniers chiffres sont présents.

ROUTAGE BANCAIRE

Numéro d'acheminement du compte bancaire aux États-Unis. Ils comportent généralement neuf chiffres, mais Amazon Comprehend reconnaît également les numéros de routage lorsque seuls les quatre derniers chiffres sont présents.

NUMÉRO_PASSEPORT

Un numéro de passeport américain. Les numéros de passeport comprennent de six à neuf caractères alphanumériques.

NUMÉRO_IDENTIFICATION_FISCAL INDIVIDUEL AMÉRICAIN

Un numéro d'identification de contribuable individuel (ITIN) américain est un numéro à neuf chiffres commençant par un « 9 » et contenant un « 7 » ou un « 8 » comme quatrième chiffre. Un ITIN peut être formaté avec un espace ou un tiret après le troisième et le quatrième chiffre.

SSN

Un numéro de sécurité sociale américain (SSN) est un numéro à neuf chiffres délivré aux citoyens américains, aux résidents permanents et aux résidents actifs temporaires. Amazon Comprehend reconnaît également les numéros de sécurité sociale lorsque seuls les quatre derniers chiffres sont présents.

Étiquetage des entités PII

Lorsque vous exécutez la détection des informations personnelles, Amazon Comprehend renvoie les étiquettes des types d'entités à informations personnelles identifiés. Par exemple, si vous envoyez le texte d'entrée suivant à Amazon Comprehend :

Bonjour Paulo Santos. Le dernier relevé de votre compte de carte de crédit 1111-0000-1111-0000 a été envoyé par la poste au 123 Any Street, Seattle, WA 98109.

La sortie inclut des étiquettes qui représentent les types d'entités PII ainsi qu'un score de confiance quant à la précision. Dans ce cas, les textes du document « Paul Santos », « 1111-0000-1111-0000 » et « 123 Any Street, Seattle, WA 98109 » génèrent les étiquettes`NAME`, `CREDIT_DEBIT_NUMBER` respectivement sous forme de types d'entités PII. `ADDRESS` Pour plus d'informations sur les types d'entités pris en charge, consultezTypes d'entités universelles PII.

Amazon Comprehend fournit les informations suivantes pour chaque étiquette :

- Le nom d'étiquette du type d'entité PII.
- Un score qui estime la probabilité que le texte détecté soit étiqueté comme un type d'entité PII.

L'exemple de texte d'entrée ci-dessus génère la sortie JSON suivante.

```
{
    "Labels": [
        {
            "Name": "NAME",
            "Score": 0.9149109721183777
        },
        {
            "Name": "CREDIT_DEBIT_NUMBER",
            "Score": 0.5698626637458801
        }
         {
            "Name": "ADDRESS",
            "Score": 0.9951046109199524
        }
    ]
}
```

Analyse en temps réel des informations personnelles (console)

Vous pouvez utiliser la console pour exécuter la détection en temps réel des informations personnelles d'un document texte. La taille maximale du texte est de 100 kilo-octets de caractères codés en UTF-8. La console affiche les résultats afin que vous puissiez consulter l'analyse.

Exécutez une analyse en temps réel de la détection des informations personnelles à l'aide du modèle intégré

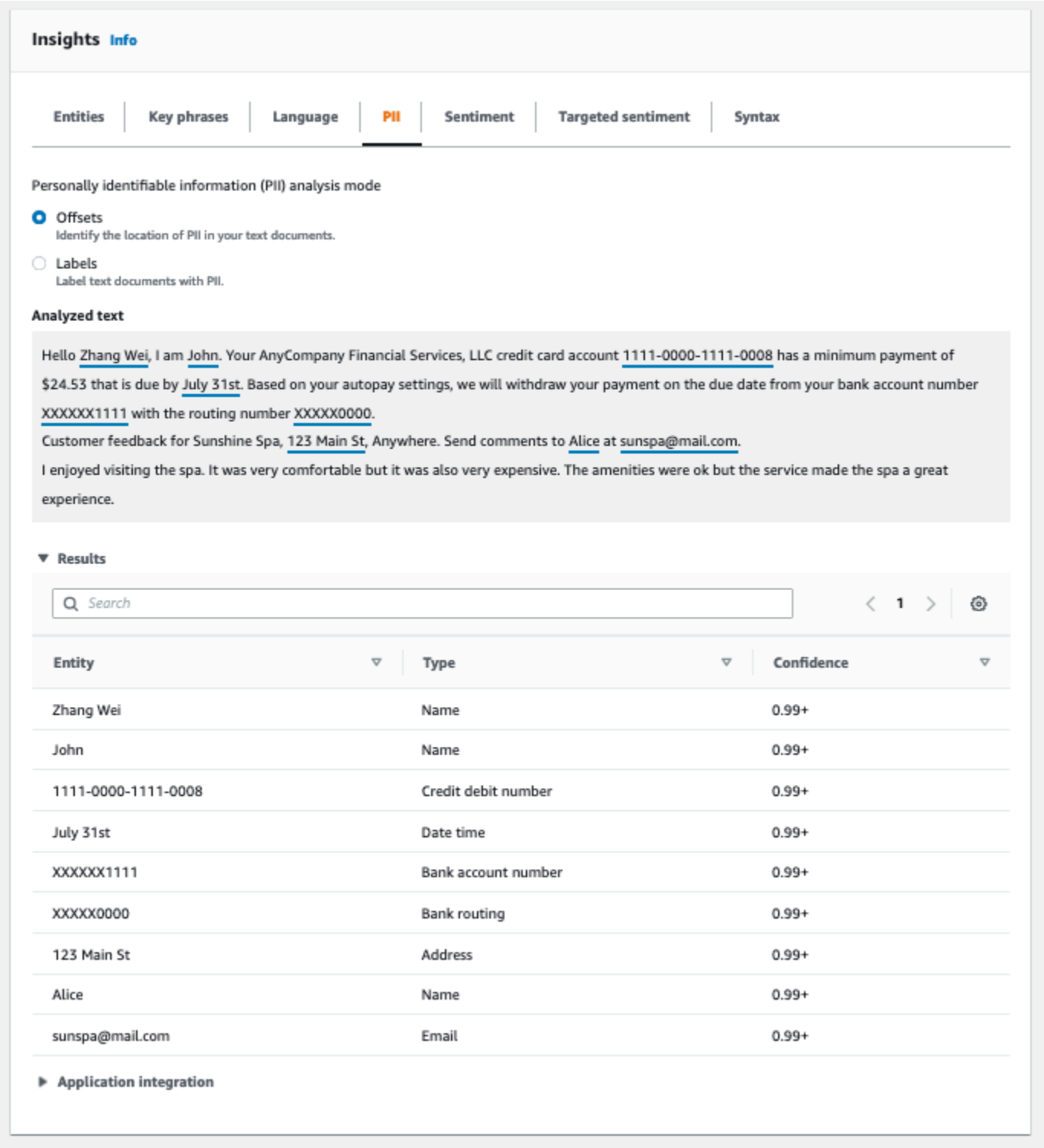
1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, sélectionnez Analyse en temps réel.
3. Sous Type d'entrée, choisissez Intégré pour le type d'analyse.
4. Entrez le texte que vous souhaitez analyser.
5. Choisissez Analyser. La console affiche les résultats de l'analyse du texte dans le panneau Insights. L'onglet PII répertorie les entités PII détectées dans le texte saisi.

Dans le panneau Insights, l'onglet PII affiche les résultats de deux modes d'analyse :

- Offsets : identifie l'emplacement des informations personnelles dans le document texte.
- Étiquettes : identifie les étiquettes des types d'entités PII identifiés.

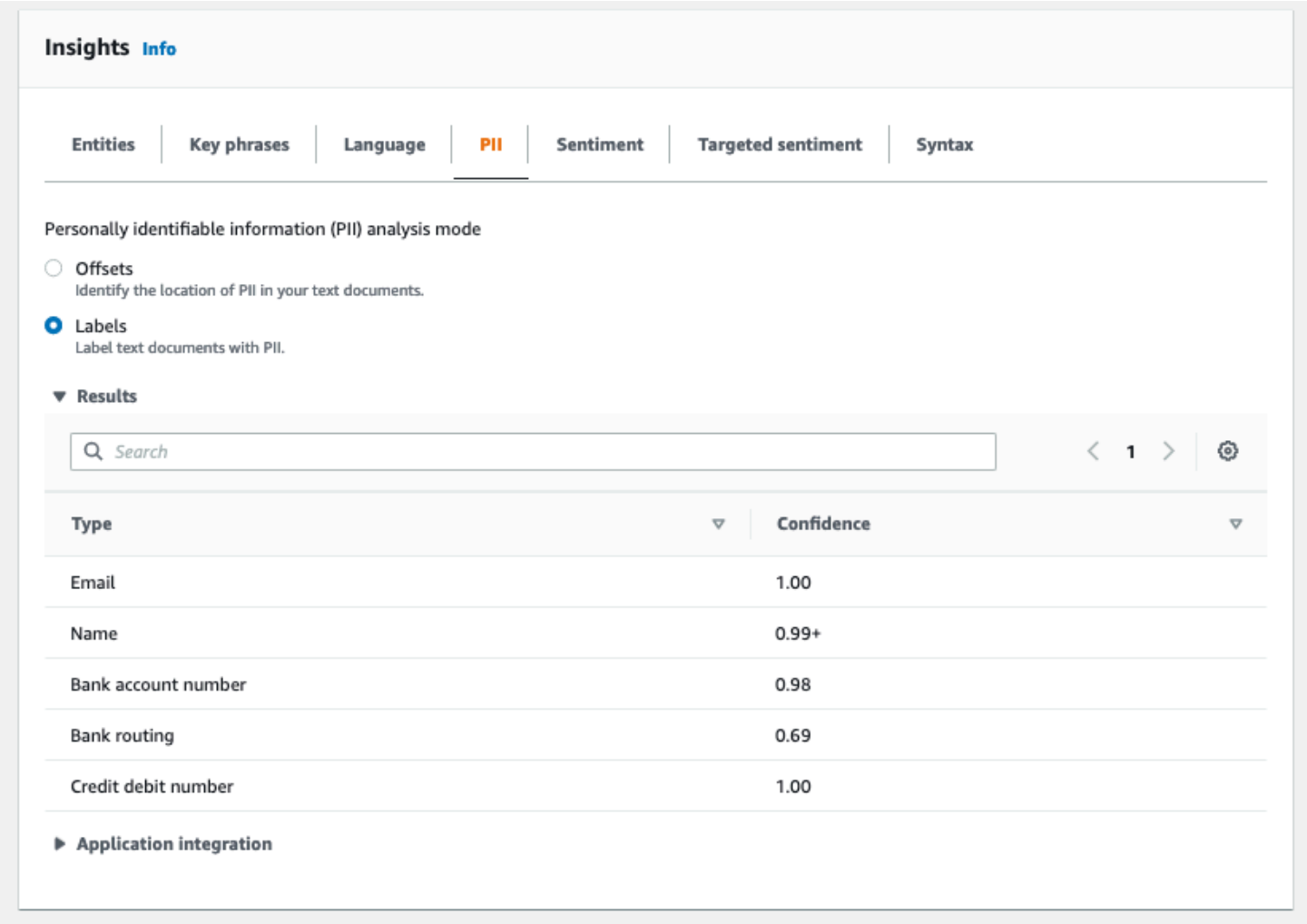
Offsets

Le mode d'analyse des décalages identifie l'emplacement des informations personnelles dans vos documents texte. Pour de plus amples informations, veuillez consulter Localiser les entités PII.



Étiquettes

Le mode d'analyse des étiquettes renvoie les étiquettes des types d'entités PII identifiés. Pour de plus amples informations, veuillez consulter Étiquetage des entités PII.



Tâches d'analyse asynchrones PII (console)

Vous pouvez utiliser la console pour créer des tâches d'analyse asynchrones afin de détecter les entités PII. Pour plus d'informations sur les types d'entités PII, consultezDétection des entités PII.

Pour créer une tâche d'analyse

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Tâches d'analyse, puis sélectionnez Créer une tâche.
3. Sous Paramètres de la tâche, attribuez un nom unique à la tâche d'analyse.
4. Pour le type d'analyse, sélectionnez Informations personnelles identifiables (PII).
5. Dans Langue, choisissez l'une des langues prises en charge (anglais ou espagnol).
6. Dans Mode sortie, sélectionnez l'une des options suivantes :

- Offsets : le résultat de la tâche renvoie l'emplacement de chaque entité PII.
- Rédactions — La sortie de la tâche renvoie une copie du texte d'entrée avec chaque entrée PII expurgée.

7. (Facultatif) Si vous choisissez Redactions comme mode de sortie, vous pouvez sélectionner les types d'entités PII à expurger.
8. Sous Données d'entrée, spécifiez l'emplacement des documents d'entrée dans Amazon S3 :
   - Pour analyser vos propres documents, choisissez Mes documents, puis Browse S3 pour indiquer le chemin d'accès au compartiment ou au dossier contenant vos fichiers.
   - Pour analyser les échantillons fournis par Amazon Comprehend, sélectionnez Example documents. Dans ce cas, Amazon Comprehend utilise un bucket géré par AWS, et vous n'en spécifiez pas l'emplacement.
9. (Facultatif) Dans Format d'entrée, spécifiez l'un des formats suivants pour vos fichiers d'entrée :
   - Un document par fichier — Chaque fichier contient un document d'entrée. C'est la meilleure solution pour les collections de documents volumineux.
   - Un document par ligne — L'entrée est un ou plusieurs fichiers. Chaque ligne d'un fichier est considérée comme un document. C'est la meilleure solution pour les documents courts, tels que les publications sur les réseaux sociaux. Chaque ligne doit se terminer par une ligne d'alimentation (LF,\n), un retour en chariot (CR, \ r), ou les deux (CRLF, \ r\n). Vous ne pouvez pas utiliser le séparateur de ligne UTF-8 (u+2028) pour terminer une ligne.
10. Sous Données de sortie, choisissez Browse S3. Choisissez le compartiment ou le dossier Amazon S3 dans lequel vous souhaitez qu'Amazon Comprehend écrive les données de sortie produites par l'analyse.
11. (Facultatif) Pour chiffrer le résultat de sortie de votre tâche, choisissez Chiffrement. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte :
    - Si vous utilisez une clé associée au compte actuel, choisissez l'alias de clé ou l'ID pour l'ID de clé KMS.
    - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'alias ou de l'ID de clé sous ID de clé KMS.

> **Note**
>
> Pour plus d'informations sur la création et l'utilisation des clés KMS et le chiffrement associé, consultez la section Service de gestion des clés (KMS).

12. Sous Autorisations d'accès, fournissez un rôle IAM qui :

    - Accorde un accès en lecture à l'emplacement Amazon S3 de vos documents d'entrée.
    - Accorde un accès en écriture à l'emplacement Amazon S3 de vos documents de sortie.
    - Inclut une politique de confiance qui permet au directeur du `comprehend.amazonaws.com` service d'assumer le rôle et d'obtenir ses autorisations.

    Si vous ne possédez pas encore de rôle IAM doté de ces autorisations et d'une politique de confiance appropriée, choisissez Créer un rôle IAM pour en créer un.
13. Lorsque vous avez fini de remplir le formulaire, choisissez Créer une tâche pour créer et démarrer la tâche de détection de sujets.

La nouvelle tâche apparaît dans la liste des tâches avec le champ d'état indiquant le statut de la tâche. Le champ peut être `IN_PROGRESS` destiné à une tâche en cours de traitement, `COMPLETED` à une tâche terminée avec succès `FAILED` ou à une tâche comportant une erreur. Vous pouvez cliquer sur une tâche pour obtenir plus d'informations sur la tâche, y compris les messages d'erreur éventuels.

Lorsque la tâche est terminée, Amazon Comprehend stocke les résultats de l'analyse dans l'emplacement Amazon S3 en sortie que vous avez spécifié pour la tâche. Pour une description des résultats de l'analyse, voirDétection des entités PII.

Analyse en temps réel des informations personnelles (API)

Amazon Comprehend fournit des opérations d'API synchrones en temps réel pour analyser les informations personnelles identifiables (PII) contenues dans un document.

Rubriques

- Localisation des entités PII en temps réel (API)
- Étiquetage des entités PII en temps réel (API)

Localisation des entités PII en temps réel (API)

Pour localiser les informations personnelles dans un seul document, vous pouvez utiliser l'opération Amazon DetectPiiEntitiesComprehend. Le texte saisi peut inclure jusqu'à 100 kilo-octets de caractères codés en UTF-8. Les langues prises en charge incluent l'anglais et l'espagnol.

Localisation des informations personnelles à l'aide de (CLI)

L'exemple suivant utilise l'`DetectPiiEntities`opération avec le AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend detect-pii-entities \
  --text "Hello Paul Santos. The latest statement for your credit card \
  account 1111-0000-1111-0000 was mailed to 123 Any Street, Seattle, WA \
   98109." \
  --language-code en
```

Amazon Comprehend répond comme suit :

```
{
      "Entities": [
          {
              "Score": 0.9999669790267944,
              "Type": "NAME",
              "BeginOffset": 6,
              "EndOffset": 18
          },
          {
              "Score": 0.8905550241470337,
              "Type": "CREDIT_DEBIT_NUMBER",
              "BeginOffset": 69,
              "EndOffset": 88
          },
          {
              "Score": 0.9999889731407166,
              "Type": "ADDRESS",
              "BeginOffset": 103,
              "EndOffset": 138
          }
```

```
      ]
  }
```

Étiquetage des entités PII en temps réel (API)

Vous pouvez utiliser les opérations d'API synchrones en temps réel pour renvoyer les étiquettes des types d'entités PII identifiés. Pour de plus amples informations, veuillez consulter Étiquetage des entités PII.

Étiquetage des entités PII (CLI)

L'exemple suivant utilise l'`ContainsPiiEntities`opération avec le AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend contains-pii-entities \
--text "Hello Paul Santos. The latest statement for your credit card \
account 1111-0000-1111-0000 was mailed to 123 Any Street, Seattle, WA \
 98109." \
--language-code en
```

Amazon Comprehend répond comme suit :

```
{
    "Labels": [
        {
            "Name": "NAME",
            "Score": 0.9149109721183777
        },
        {
            "Name": "CREDIT_DEBIT_NUMBER",
            "Score": 0.8905550241470337
        }
         {
            "Name": "ADDRESS",
            "Score": 0.9951046109199524
        }
    ]
}
```

Tâches d'analyse asynchrones PII (API)

Analyse asynchrone des PII (API)

Vous pouvez utiliser des opérations d'API asynchrones pour créer des tâches d'analyse afin de localiser ou de supprimer des entités PII. Pour plus d'informations sur les types d'entités PII, consultezDétection des entités PII.

Rubriques

- Localisation d'entités PII avec des tâches asynchrones (API)
- Rédaction d'entités PII avec des tâches asynchrones (API)

Localisation d'entités PII avec des tâches asynchrones (API)

Exécutez un traitement par lots asynchrone pour localiser les informations personnelles dans un ensemble de documents. Pour exécuter la tâche, téléchargez vos documents sur Amazon S3 et soumettez une StartPiiEntitiesDetectionJobdemande.

Rubriques

- Avant de commencer
- Paramètres d'entrée
- Méthodes Async Job
- Format de fichier de sortie
- Analyse asynchrone à l'aide du AWS Command Line Interface

Avant de commencer

Avant de commencer, assurez-vous que vous disposez des éléments suivants :

- Compartiments d'entrée et de sortie : identifiez les compartiments Amazon S3 que vous souhaitez utiliser pour les fichiers d'entrée et de sortie. Les buckets doivent se trouver dans la même région que l'API que vous appelez.
- Rôle de service IAM : vous devez disposer d'un rôle de service IAM autorisé à accéder à vos compartiments d'entrée et de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Paramètres d'entrée

Dans votre demande, incluez les paramètres obligatoires suivants :

- `InputDataConfig`— Fournissez une InputDataConfigdéfinition pour votre demande, qui inclut les propriétés d'entrée de la tâche. Pour le `S3Uri` paramètre, spécifiez l'emplacement Amazon S3 de vos documents d'entrée.
- `OutputDataConfig`— Fournissez une OutputDataConfigdéfinition pour votre demande, qui inclut les propriétés de sortie de la tâche. Pour le `S3Uri` paramètre, spécifiez l'emplacement Amazon S3 où Amazon Comprehend écrit les résultats de son analyse.
- `DataAccessRoleArn`— Fournissez le nom de ressource Amazon (ARN) d'un AWS Identity and Access Management rôle. Ce rôle doit accorder à Amazon Comprehend un accès en lecture à vos données d'entrée et un accès en écriture à votre emplacement de sortie dans Amazon S3. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.
- `Mode`— Définissez ce paramètre sur`ONLY_OFFSETS`. Avec ce paramètre, la sortie fournit les décalages de caractères qui localisent chaque entité PII dans le texte d'entrée. Le résultat inclut également les scores de confiance et les types d'entités PII.
- `LanguageCode`— Définissez ce paramètre sur `en` ou`es`. Amazon Comprehend prend en charge la détection des informations personnelles dans des textes en anglais ou en espagnol.

Méthodes Async Job

`StartPiiEntitiesDetectionJob`Renvoie un identifiant de tâche, afin que vous puissiez suivre la progression de la tâche et récupérer le statut de la tâche une fois celle-ci terminée.

Pour suivre la progression d'une tâche d'analyse, fournissez l'identifiant de la tâche à l'DescribePiiEntitiesDetectionJobopération. Le formulaire de réponse `DescribePiiEntitiesDetectionJob` contient le `JobStatus` champ indiquant le statut actuel de la tâche. Un emploi réussi passe par les états suivants :

SOUMIS -> IN_PROGRESS -> TERMINÉ.

Une fois qu'une tâche d'analyse `JobStatus` est terminée (TERMINÉE, ÉCHOUÉE ou ARRÊTÉE), utilisez-la `DescribePiiEntitiesDetectionJob` pour obtenir l'emplacement des résultats. Si le statut de la tâche est le suivant`COMPLETED`, la réponse inclut un `OutputDataConfig` champ contenant l'emplacement du fichier de sortie sur Amazon S3.

Pour plus de détails sur les étapes à suivre pour l'analyse asynchrone d'Amazon Comprehend, consultez. Traitement par lots asynchrone

Format de fichier de sortie

Le fichier de sortie utilise le nom du fichier d'entrée, avec .out ajouté à la fin. Il contient les résultats de l'analyse.

Voici un exemple de fichier de sortie issu d'une tâche d'analyse qui a détecté des entités PII dans des documents. Le format de l'entrée est d'un document par ligne.

```
{
  "Entities": [
    {
      "Type": "NAME",
      "BeginOffset": 40,
      "EndOffset": 69,
      "Score": 0.999995
    },
    {
      "Type": "ADDRESS",
      "BeginOffset": 247,
      "EndOffset": 253,
      "Score": 0.998828
    },
    {
      "Type": "BANK_ACCOUNT_NUMBER",
      "BeginOffset": 406,
      "EndOffset": 411,
      "Score": 0.693283
    }
  ],
  "File": "doc.txt",
  "Line": 0
},
{
  "Entities": [
    {
      "Type": "SSN",
      "BeginOffset": 1114,
      "EndOffset": 1124,
      "Score": 0.999999
    },
```

```
    {
      "Type": "EMAIL",
      "BeginOffset": 3742,
      "EndOffset": 3775,
      "Score": 0.999993
    },
    {
      "Type": "PIN",
      "BeginOffset": 4098,
      "EndOffset": 4102,
      "Score": 0.999995
    }
  ],
  "File": "doc.txt",
  "Line": 1
 }
```

Voici un exemple de sortie d'une analyse où le format de l'entrée est d'un document par fichier.

```
{
  "Entities": [
    {
      "Type": "NAME",
      "BeginOffset": 40,
      "EndOffset": 69,
      "Score": 0.999995
    },
    {
      "Type": "ADDRESS",
      "BeginOffset": 247,
      "EndOffset": 253,
      "Score": 0.998828
    },
    {
      "Type": "BANK_ROUTING",
      "BeginOffset": 279,
      "EndOffset": 289,
      "Score": 0.999999
    }
  ],
  "File": "doc.txt"
}
```

Analyse asynchrone à l'aide du AWS Command Line Interface

L'exemple suivant utilise l'`StartPiiEntitiesDetectionJob`opération avec le AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend start-pii-entities-detection-job \
    --region region \
    --job-name job name \
    --cli-input-json file://path to JSON input file
```

Pour le `cli-input-json` paramètre, vous fournissez le chemin d'accès à un fichier JSON contenant les données de demande, comme illustré dans l'exemple suivant.

```
{
  "InputDataConfig": {
      "S3Uri": "s3://input bucket/input path",
      "InputFormat": "ONE_DOC_PER_LINE"
  },
  "OutputDataConfig": {
      "S3Uri": "s3://output bucket/output path"
  },
  "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"
  "LanguageCode": "en",
  "Mode": "ONLY_OFFSETS"
}
```

Si la demande de démarrage de la tâche de détection des événements est réussie, vous recevrez une réponse similaire à la suivante :

```
{
  "JobId": "5d2fbe6e...e2c"
  "JobArn":  "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-
job/5d2fbe6e...e2c"
  "JobStatus": "SUBMITTED",
}
```

Vous pouvez utiliser cette DescribeEventsDetectionJobopération pour obtenir le statut d'une tâche existante. Si la demande de démarrage de la tâche de détection des événements est réussie, vous recevrez une réponse similaire à la suivante :

```
aws comprehend describe-pii-entities-detection-job \
    --region region \
    --job-id job ID
```

Lorsque le travail est terminé avec succès, vous recevez une réponse similaire à la suivante :

```
{
    "PiiEntitiesDetectionJobProperties": {
  "JobId": "5d2fbe6e...e2c"
  "JobArn":  "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-
job/5d2fbe6e...e2c"
  "JobName": "piiCLItest3",
  "JobStatus": "COMPLETED",
  "SubmitTime": "2022-05-05T14:54:06.169000-07:00",
  "EndTime": "2022-05-05T15:00:17.007000-07:00",
  "InputDataConfig": {
       (identical to the input data that you provided with the request)
    }
}
```

Rédaction d'entités PII avec des tâches asynchrones (API)

Pour supprimer les entités PII de votre texte, vous devez démarrer un traitement par lots asynchrone. Pour exécuter la tâche, téléchargez vos documents sur Amazon S3 et soumettez une StartPiiEntitiesDetectionJobdemande.

Rubriques

- Avant de commencer
- Paramètres d'entrée
- Format de fichier de sortie
- Rédaction des informations personnelles à l'aide du AWS Command Line Interface

Avant de commencer

Avant de commencer, assurez-vous que vous disposez des éléments suivants :

- Compartiments d'entrée et de sortie : identifiez les compartiments Amazon S3 que vous souhaitez utiliser pour les fichiers d'entrée et de sortie. Les buckets doivent se trouver dans la même région que l'API que vous appelez.
- Rôle de service IAM : vous devez disposer d'un rôle de service IAM autorisé à accéder à vos compartiments d'entrée et de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Paramètres d'entrée

Dans votre demande, incluez les paramètres obligatoires suivants :

- `InputDataConfig`— Fournissez une InputDataConfigdéfinition pour votre demande, qui inclut les propriétés d'entrée de la tâche. Pour le `S3Uri` paramètre, spécifiez l'emplacement Amazon S3 de vos documents d'entrée.
- `OutputDataConfig`— Fournissez une OutputDataConfigdéfinition pour votre demande, qui inclut les propriétés de sortie de la tâche. Pour le `S3Uri` paramètre, spécifiez l'emplacement Amazon S3 où Amazon Comprehend écrit les résultats de son analyse.
- `DataAccessRoleArn`— Fournissez le nom de ressource Amazon (ARN) d'un AWS Identity and Access Management rôle. Ce rôle doit accorder à Amazon Comprehend un accès en lecture à vos données d'entrée et un accès en écriture à votre emplacement de sortie dans Amazon S3. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.
- `Mode`— Définissez ce paramètre sur`ONLY_REDACTION`. Avec ce paramètre, Amazon Comprehend écrit une copie de vos documents d'entrée sur l'emplacement de sortie dans Amazon S3. Dans cette copie, chaque entité PII est expurgée.
- `RedactionConfig`— Fournissez une RedactionConfigdéfinition pour votre demande, qui inclut les paramètres de configuration pour la rédaction. Spécifiez les types de PII à supprimer, et précisez si chaque entité PII est remplacée par le nom de son type ou par un caractère de votre choix :
  - Spécifiez les types d'entités PII à supprimer dans le `PiiEntityTypes` tableau. Pour supprimer tous les types d'entités, définissez la valeur du tableau sur`["ALL"]`.
  - Pour remplacer chaque entité PII par son type, définissez le `MaskMode` paramètre sur. `REPLACE_WITH_PII_ENTITY_TYPE` Par exemple, avec ce paramètre, l'entité PII « Jane Doe » est remplacée par « [NAME] ».

- Pour remplacer les caractères de chaque entité PII par un caractère de votre choix, définissez le `MaskMode` paramètre sur et définissez le `MaskCharacter` paramètre sur le caractère de remplacement. `MASK` Ne fournissez qu'un seul caractère. Les caractères valides sont ! , #, $, %, &, * et @. Par exemple, avec ce paramètre, l'entité PII « Jane Doe » peut être remplacée par « **** *** »
- `LanguageCode`— Définissez ce paramètre sur `en` ou`es`. Amazon Comprehend prend en charge la détection des informations personnelles dans des textes en anglais ou en espagnol.

Format de fichier de sortie

L'exemple suivant montre les fichiers d'entrée et de sortie d'une tâche d'analyse qui expédie des informations personnelles. Le format de l'entrée est d'un document par ligne.

```
{
Managing Your Accounts Primary Branch Canton John Doe Phone Number 443-573-4800 123
 Main StreetBaltimore, MD 21224
Online Banking HowardBank.com  Telephone 1-877-527-2703 Bank 3301 Boston Street,
 Baltimore, MD 21224
```

La tâche d'analyse visant à supprimer ce fichier d'entrée produit le fichier de sortie suivant.

```
{
Managing Your Accounts Primary Branch ****** ******** Phone Number ************
 ********************************
Online Banking **************  Telephone ************** Bank
 ************************************
 }
```

Rédaction des informations personnelles à l'aide du AWS Command Line Interface

L'exemple suivant utilise l'`StartPiiEntitiesDetectionJob`opération avec le AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend start-pii-entities-detection-job \
    --region region \
```

```
    --job-name job name \
    --cli-input-json file://path to JSON input file
```

Pour le `cli-input-json` paramètre, vous fournissez le chemin d'accès à un fichier JSON contenant les données de demande, comme illustré dans l'exemple suivant.

```
{
    "InputDataConfig": {
        "S3Uri": "s3://input bucket/input path",
        "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
        "S3Uri": "s3://output bucket/output path"
    },
    "DataAccessRoleArn": "arn:aws:iam::account ID:role/data access role"
    "LanguageCode": "en",
    "Mode": "ONLY_REDACTION"
    "RedactionConfig": {
        "MaskCharacter": "*",
        "MaskMode": "MASK",
        "PiiEntityTypes": ["ALL"]
    }
}
```

Si la demande de démarrage de la tâche de détection des événements est réussie, vous recevrez une réponse similaire à la suivante :

```
{
  "JobId": "7c4fbe6e...e5b"
  "JobArn":  "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-
job/7c4fbe6e...e5b"
  "JobStatus": "SUBMITTED",
}
```

Vous pouvez utiliser cette DescribeEventsDetectionJobopération pour obtenir le statut d'une tâche existante.

```
aws comprehend describe-pii-entities-detection-job \
    --region region \
    --job-id job ID
```

Lorsque le travail est terminé avec succès, vous recevez une réponse similaire à la suivante :

```
{
  "PiiEntitiesDetectionJobProperties": {
     "JobId": "7c4fbe6e...e5b"
     "JobArn":  "arn:aws:comprehend:us-west-2:123456789012:pii-entities-detection-
job/7c4fbe6e...e5b"
     "JobName": "piiCLIredtest1",
     "JobStatus": "COMPLETED",
     "SubmitTime": "2022-05-05T14:54:06.169000-07:00",
     "EndTime": "2022-05-05T15:00:17.007000-07:00",
     "InputDataConfig": {
        (identical to the input data that you provided with the request)
  }
}
```

Traitement des documents

Amazon Comprehend prend en charge le traitement des documents en une étape pour une classification personnalisée et une reconnaissance d'entités personnalisées. Par exemple, vous pouvez saisir un mélange de documents en texte brut et de documents semi-structurés (tels que des documents PDF, des documents Microsoft Word et des images) dans une tâche d'analyse personnalisée.

Pour les fichiers d'entrée qui nécessitent une extraction de texte, Amazon Comprehend effectue automatiquement l'extraction du texte avant d'exécuter l'analyse. Pour extraire le contenu du texte, Amazon Comprehend utilise un analyseur interne pour les documents semi-structurés natifs et utilise Amazon Textract pour les images et les documents numérisés. APIs

Le traitement des documents par Amazon Comprehend est disponible dans chacun des Amazon Régions prises en charge Comprehend, à l'exception de la région Asie-Pacifique (Tokyo) AWS GovCloud et (ouest des États-Unis) qui ne prennent en charge que les modèles en texte brut pour une classification personnalisée.

Les rubriques suivantes fournissent des informations sur les types de documents d'entrée pris en charge par Amazon Comprehend pour les analyses personnalisées.

Rubriques

- Entrées pour une analyse personnalisée en temps réel
- Entrées pour une analyse personnalisée asynchrone
- Configuration des options d'extraction de texte
- Bonnes pratiques pour les images

Entrées pour une analyse personnalisée en temps réel

L'analyse en temps réel à l'aide de modèles personnalisés prend un seul document en entrée. Les rubriques suivantes décrivent les types de documents d'entrée que vous pouvez utiliser.

Rubriques

- Documents en texte brut
- Documents semi-structurés
- Fichiers image et fichiers PDF numérisés

- Sortie Amazon Textract
- Tailles de document maximales pour une analyse en temps réel
- Erreurs dans les documents semi-structurés

Documents en texte brut

Fournissez le document d'entrée sous forme de texte au format UTF-8.

Documents semi-structurés

Les documents semi-structurés incluent les documents PDF natifs et les documents Word.

Par défaut, l'analyse personnalisée en temps réel utilise l'analyseur Amazon Comprehend pour extraire le texte des fichiers Word et des fichiers PDF numériques. Pour les fichiers PDF, vous pouvez annuler cette valeur par défaut et utiliser Amazon Textract pour extraire le texte. Consultez Configuration des options d'extraction de texte.

Fichiers image et fichiers PDF numérisés

Les types d'image pris en charge sont les suivants : JPEG, PNG et TIFF.

Par défaut, la reconnaissance d'entité personnalisée utilise l'opération d'`DetectDocumentText`API Amazon Textract pour extraire le texte des fichiers image et des fichiers PDF numérisés. Vous pouvez remplacer cette valeur par défaut pour utiliser l'opération `AnalyzeDocument` API à la place. Consultez Configuration des options d'extraction de texte.

Sortie Amazon Textract

Vous pouvez fournir la sortie JSON de l'`DetectDocumentText`API ou `AnalyzeDocument` de l'API Amazon Textract en tant qu'entrée pour les opérations d'API en temps réel pour une classification personnalisée et une reconnaissance d'entité personnalisée. Amazon Comprehend prend en charge ce type d'entrée pour les opérations d'API en temps réel, mais pas pour la console.

Tailles de document maximales pour une analyse en temps réel

Pour tous les types de documents d'entrée, le maximum du fichier d'entrée est d'une page, avec un maximum de 10 000 caractères.

Le tableau suivant indique les tailles de fichier maximales pour les documents d'entrée.

Type de fichier	Taille maximale (API)	Taille maximale (console)
Documents texte UTF-8	10 Ko	10 Ko
Documents au format PDF	10 Mo	5 Mo
Documents Word	10 Mo	1 Mo
Fichiers image	10 Mo	5 Mo
Textract les fichiers de sortie	1 Mo	N/A

Erreurs dans les documents semi-structurés

L'opération ClassifyDocumentou DetectEntitiesAPI peut rencontrer des erreurs au niveau du document ou au niveau de la page lors de l'extraction du texte d'un document semi-structuré ou d'un fichier image.

Erreurs au niveau de la page

Si l'opération ClassifyDocumentou DetectEntitiesAPI rencontre des erreurs lors du traitement d'une page dans le document d'entrée, la réponse de l'API inclut une entrée dans la liste des erreurs pour chaque erreur.

L'entrée `ErrorCode` de la liste d'erreurs contient l'une des valeurs suivantes :

- TEXTRACT_BAD_PAGE — Amazon Textract ne peut pas lire la page. Pour plus d'informations sur les limites de pages dans Amazon Textract, consultez la section Quotas de pages dans Amazon Textract.
- TEXTRACT_PROVISIONED_THROUGHPUT_EXCEDED — Le nombre de demandes a dépassé votre limite de débit. Pour plus d'informations sur les quotas de débit dans Amazon Textract, consultez la section Quotas par défaut dans Amazon Textract.
- PAGE_CHARACTERS_EXCEDED — Trop de caractères de texte sur la page (10 000 caractères maximum).
- PAGE_SIZE_EXCEDED — La taille de page maximale est de 10 Mo.
- INTERNAL_SERVER_ERROR — La demande a rencontré un problème de service. Réessayez la demande d'API.

Erreurs au niveau du document

Si l'opération ClassifyDocumentou DetectEntitiesAPI détecte une erreur au niveau du document dans votre document d'entrée, l'API renvoie une réponse d'`InvalidRequestException`erreur.

Dans la réponse d'erreur, le Reason champ contient la valeur`INVALID_DOCUMENT`.

Le Detail champ contient l'une des valeurs suivantes :

- DOCUMENT_SIZE_EXCEDED — La taille du document est trop grande. Vérifiez la taille de votre fichier et soumettez à nouveau la demande.
- UNSUPPORTED_DOC_TYPE — Le type de document n'est pas pris en charge. Vérifiez le type de fichier et soumettez à nouveau la demande.
- PAGE_LIMIT_EXCEDED — Le document contient trop de pages. Vérifiez le nombre de pages de votre fichier et soumettez à nouveau la demande.
- TEXTRACT_ACCESS_DENIED_EXCEPTION — Accès refusé à Amazon Textract. Vérifiez que votre compte est autorisé à utiliser les opérations Amazon Textract DetectDocumentTextet AnalyzeDocumentAPI, puis soumettez à nouveau la demande.

Entrées pour une analyse personnalisée asynchrone

Vous pouvez saisir plusieurs documents dans une tâche d'analyse asynchrone personnalisée. Les rubriques suivantes décrivent les types de documents d'entrée que vous pouvez utiliser. La taille maximale du fichier varie en fonction du type de document d'entrée.

Rubriques

- Documents en texte brut
- Documents semi-structurés
- Fichiers image et fichiers PDF numérisés
- Fichiers JSON de sortie Amazon Textract

Documents en texte brut

Fournissez tous les documents d'entrée en texte brut sous forme de texte au format UTF-8. Le tableau suivant répertorie les tailles de fichier maximales et d'autres directives.

> **Note**
>
> Ces limites s'appliquent lorsque tous les fichiers d'entrée sont en texte brut.

Description	Quot/ligne directrice
Taille de fichier maximale pour un document par format de fichier (classification personnalisée)	1 octet—10 Mo
Taille du document (reconnaissance d'entité personnalisée)	1 octet—1 Mo
Nombre maximum de fichiers, un document par fichier	1 000 000
Nombre maximum de lignes, un document par ligne (pour tous les fichiers demandés)	1 000 000
Taille du corpus du document (tous les documents en texte brut combinés)	1 octet—5 Go

Documents semi-structurés

Les documents semi-structurés incluent les documents PDF natifs et les documents Word.

Le tableau suivant répertorie les tailles de fichier maximales et d'autres directives.

Description	Quot/ligne directrice
Taille du document (PDF)	1 octet—50 Mo
Taille du document (Docx)	1 octet—5 Mo
Nombre maximum de fichiers	500
Nombre maximum de pages pour un fichier PDF ou Docx	100
Taille du corpus du document après extraction du texte (texte brut, tous fichiers confondus)	1 octet—5 Go

Par défaut, l'analyse personnalisée utilise l'analyseur Amazon Comprehend pour extraire le texte des fichiers Word et des fichiers PDF numériques. Pour les fichiers PDF, vous pouvez annuler cette valeur par défaut et utiliser Amazon Textract pour extraire le texte. Consultez Configuration des options d'extraction de texte.

Fichiers image et fichiers PDF numérisés

L'analyse personnalisée prend en charge les images JPEG, PNG et TIFF.

Le tableau suivant répertorie les tailles de fichier maximales pour les images. Les fichiers PDF numérisés sont soumis aux mêmes tailles maximales que les fichiers PDF natifs.

Description	Quot/ligne directrice
Taille de l'image (JPG ou PNG)	1 octet—10 Mo
Taille de l'image (TIFF)	1 octet à 10 Mo. Une page au maximum.

Pour plus d'informations sur les images, voirBonnes pratiques pour les images.

Par défaut, Amazon Comprehend utilise l'opération d'API Amazon `DetectDocumentText` Textract pour extraire le texte des fichiers image et des fichiers PDF numérisés. Vous pouvez remplacer cette valeur par défaut pour utiliser l'opération `AnalyzeDocument` API à la place. Consultez Configuration des options d'extraction de texte.

Fichiers JSON de sortie Amazon Textract

Pour une reconnaissance d'entité personnalisée, mais pas pour une classification personnalisée, vous pouvez fournir le fichier de sortie de l'opération d'`AnalyzeDocument`API Amazon Textract en tant qu'entrée pour les tâches d'analyse.

Configuration des options d'extraction de texte

Par défaut, Amazon Comprehend exécute les actions suivantes pour extraire le texte d'un fichier, en fonction du type de fichier d'entrée :

- Fichiers Word : l'analyseur Amazon Comprehend extrait le texte.

- Fichiers PDF numériques — L'analyseur Amazon Comprehend extrait le texte.
- Fichiers image et fichiers PDF numérisés : Amazon Comprehend utilise l'API Amazon `DetectDocumentText` Textract pour extraire le texte.

Pour les fichiers image et les fichiers PDF, vous pouvez utiliser le `DocumentReaderConfig` paramètre pour annuler ces actions d'extraction par défaut. Ce paramètre est disponible lorsque vous utilisez la console ou l'API Amazon Comprehend pour une analyse personnalisée en temps réel ou asynchrone.

Le `DocumentReaderConfig` paramètre contient trois champs :

- DocumentReadMode— Réglé sur `SERVICE_DEFAULT` pour qu'Amazon Comprehend exécute les actions par défaut.

  Configurez `FORCE_DOCUMENT_READ_ACTION` pour utiliser Amazon Textract pour analyser des fichiers PDF numériques.
- DocumentReadAction— Définit l'API Amazon Textract (DetectDocumentText ou AnalyzeDocument) à utiliser lorsqu'Amazon Comprehend utilise Amazon Textract pour l'extraction de texte.
- FeatureTypes— Si vous configurez DocumentReadActionpour utiliser l'opération AnalyzeDocument API, vous pouvez ajouter l'une ou les deux `FeatureTypes` (TABLES, FORMS). Ces fonctionnalités fournissent des informations supplémentaires sur les tables et les formulaires du document. Pour plus d'informations sur ces fonctionnalités, consultez Amazon Textract Document Analysis Response Objects.

Les exemples suivants montrent comment configurer `DocumentReaderConfig` pour des cas d'utilisation spécifiques :

1. Utilisez Amazon Textract pour tous les fichiers PDF.
   a. DocumentReadMode – Défini sur `FORCE_DOCUMENT_READ_ACTION`.
   b. DocumentReadAction – Défini sur `TEXTRACT_DETECT_DOCUMENT_TEXT`.
   c. FeatureTypes— Non obligatoire
2. Utilisez l'`AnalyzeDocument`API Amazon Textract pour tous les fichiers PDF et images.
   a. DocumentReadMode – Défini sur `FORCE_DOCUMENT_READ_ACTION`.
   b. DocumentReadAction – Défini sur `TEXTRACT_ANALYZE_DOCUMENT`.

   c. FeatureTypes— Réglez sur `TABLES FORMS` ou sur les deux fonctionnalités.
3. Utilisez l'`AnalyzeDocument`API Amazon Textract pour les fichiers PDF numérisés et tous les fichiers image.
   a. DocumentReadMode – Défini sur `SERVICE_DEFAULT`.
   b. DocumentReadAction – Défini sur `TEXTRACT_ANALYZE_DOCUMENT`.
   c. FeatureTypes— Réglez sur `TABLES FORMS` ou sur les deux fonctionnalités.

Pour plus d'informations sur les options d'Amazon Textract, consultez. DocumentReaderConfig

Bonnes pratiques pour les images

Lorsque vous utilisez des fichiers image pour une classification personnalisée ou une reconnaissance d'entités personnalisée, suivez les directives suivantes pour obtenir les meilleurs résultats :

- Fournissez une image de haute qualité, idéalement d'au moins 150 DPI.
- Si le fichier image utilise l'un des formats pris en charge (TIFF, JPEG ou PNG), ne le convertissez pas ou ne sous-échantillonnez pas le fichier avant de le télécharger sur Amazon S3.

Pour obtenir les meilleurs résultats lors de l'extraction de texte à partir de tableaux dans des documents, suivez les pratiques suivantes :

- Les tableaux de votre document sont séparés visuellement des éléments qui les entourent sur la page. Par exemple, le tableau n'est pas superposé à une image ou à un motif complexe.
- Le texte du tableau est à la verticale. Par exemple, le texte n'est pas pivoté par rapport aux autres textes de la page.

Lorsque vous extrayez du texte à partir de tableaux, vous pouvez obtenir des résultats incohérents dans les cas suivants :

- Les cellules de tableau fusionnées s'étendent sur plusieurs colonnes.
- Les tableaux comportent des cellules, des lignes ou des colonnes différentes des autres parties du même tableau.

Classification personnalisée

Utilisez la classification personnalisée pour organiser vos documents en catégories (classes) que vous définissez. La classification personnalisée est un processus en deux étapes. Tout d'abord, vous entraînez un modèle de classification personnalisé (également appelé classificateur) pour reconnaître les classes qui vous intéressent. Vous pouvez ensuite utiliser votre modèle pour classer autant de jeux de documents que vous le souhaitez.

Par exemple, vous pouvez classer le contenu des demandes d'assistance afin de pouvoir les acheminer vers l'équipe d'assistance appropriée. Vous pouvez également classer les e-mails reçus des clients afin de fournir des conseils en fonction du type de demande du client. Vous pouvez associer Amazon Comprehend à Amazon Transcribe pour convertir le discours en texte, puis classer les demandes provenant des appels téléphoniques d'assistance.

Vous pouvez exécuter une classification personnalisée sur un seul document de manière synchrone (en temps réel) ou démarrer une tâche asynchrone pour classer un ensemble de documents. Vous pouvez avoir plusieurs classificateurs personnalisés dans votre compte, chacun entraîné à l'aide de données différentes. La classification personnalisée prend en charge divers types de documents d'entrée, tels que le texte brut, les PDF, Word et les images.

Lorsque vous soumettez une tâche de classification, vous choisissez le modèle de classificateur à utiliser, en fonction du type de documents que vous devez analyser. Par exemple, pour analyser des documents en texte brut, vous obtenez les résultats les plus précis en utilisant un modèle que vous avez entraîné avec des documents en texte brut. Pour analyser des documents semi-structurés (tels que des PDF, Word, des images, des sorties Amazon Textract ou des fichiers numérisés), vous obtenez les résultats les plus précis en utilisant un modèle que vous avez entraîné avec des documents natifs.

Rubriques

- Préparation des données d'entraînement du classificateur
- Modèles de classification des formations
- Exécution d'analyses en temps réel
- Exécution de tâches asynchrones

Préparation des données d'entraînement du classificateur

Pour une classification personnalisée, vous entraînez le modèle en mode multi-classes ou en mode multi-étiquettes. Le mode multi-classes associe une seule classe à chaque document. Le mode multi-étiquettes associe une ou plusieurs classes à chaque document. Les formats de fichier d'entrée étant différents pour chaque mode, choisissez le mode à utiliser avant de créer les données d'entraînement.

> **Note**
>
> La console Amazon Comprehend fait référence au mode multi-classes en tant que mode à étiquette unique.

La classification personnalisée prend en charge les modèles que vous entraînez avec des documents en texte brut et les modèles que vous entraînez avec des documents natifs (tels que PDF, Word ou images). Pour plus d'informations sur les modèles de classificateur et les types de documents pris en charge, consultezModèles de classification des formations.

Pour préparer les données afin d'entraîner un modèle de classificateur personnalisé, procédez comme suit :

1. Identifiez les classes que vous souhaitez que ce classificateur analyse. Décidez quel mode utiliser (multi-classes ou multi-étiquettes).
2. Choisissez le type de modèle de classificateur, selon que le modèle est destiné à l'analyse de documents en texte brut ou de documents semi-structurés.
3. Rassemblez des exemples de documents pour chacune des classes. Pour les exigences de formation minimales, voirQuotas généraux pour le classement des documents.
4. Pour un modèle en texte brut, choisissez le format de fichier de formation à utiliser (fichier CSV ou fichier manifeste augmenté). Pour entraîner un modèle de document natif, vous devez toujours utiliser un fichier CSV.

Rubriques

- Formats de fichiers d'entraînement du classificateur
- Mode multi-classes
- Mode multi-étiquettes

Formats de fichiers d'entraînement du classificateur

Pour un modèle en texte brut, vous pouvez fournir les données d'entraînement du classificateur sous forme de fichier CSV ou de fichier manifeste augmenté que vous créez à l'aide d'AI Ground SageMaker Truth. Le fichier CSV ou le fichier manifeste augmenté inclut le texte de chaque document de formation et ses étiquettes associées.

Pour un modèle de document natif, vous fournissez les données d'entraînement de Classifier sous forme de fichier CSV. Le fichier CSV inclut le nom de fichier de chaque document de formation et ses étiquettes associées. Vous incluez les documents de formation dans le dossier de saisie Amazon S3 correspondant à la tâche de formation.

fichiers CSV

Vous fournissez des données d'entraînement étiquetées sous forme de texte codé en UTF-8 dans un fichier CSV. N'incluez pas de ligne d'en-tête. L'ajout d'une ligne d'en-tête dans votre fichier peut provoquer des erreurs d'exécution.

Pour chaque ligne du fichier CSV, la première colonne contient une ou plusieurs étiquettes de classe. Une étiquette de classe peut être n'importe quelle chaîne UTF-8 valide. Nous vous recommandons d'utiliser des noms de classe clairs dont la signification ne se chevauche pas. Le nom peut inclure des espaces blancs et peut être composé de plusieurs mots reliés par des traits de soulignement ou des traits d'union.

Ne laissez aucun espace avant ou après les virgules séparant les valeurs d'une ligne.

Le contenu exact du fichier CSV dépend du mode de classification et du type de données d'entraînement. Pour plus de détails, consultez les sections sur Mode multi-classes etMode multi-étiquettes.

Fichier manifeste augmenté

Un fichier manifeste augmenté est un ensemble de données étiqueté que vous créez à l'aide d' SageMaker AI Ground Truth. Ground Truth est un service d'étiquetage des données qui vous aide, ou qui aide le personnel que vous employez, à créer des ensembles de données de formation pour les modèles d'apprentissage automatique.

Pour plus d'informations sur Ground Truth et les résultats qu'il produit, consultez la section Utiliser SageMaker AI Ground Truth to Label Data dans le manuel Amazon SageMaker AI Developer Guide.

Les fichiers manifestes augmentés sont au format de lignes JSON. Dans ces fichiers, chaque ligne est un objet JSON complet qui contient un document de formation et ses étiquettes associées. Le contenu exact de chaque ligne dépend du mode de classification. Pour plus de détails, consultez les sections sur Mode multi-classes etMode multi-étiquettes.

Lorsque vous fournissez vos données de formation à Amazon Comprehend, vous spécifiez un ou plusieurs noms d'attributs d'étiquette. Le nombre de noms d'attributs que vous spécifiez varie selon que votre fichier manifeste augmenté est le résultat d'une tâche d'étiquetage unique ou d'une tâche d'étiquetage en chaîne.

Si votre fichier est le résultat d'une seule tâche d'étiquetage, spécifiez le nom de l'attribut d'étiquette unique issu de la tâche Ground Truth.

Si votre fichier est le résultat d'une tâche d'étiquetage en chaîne, spécifiez le nom de l'attribut d'étiquette pour une ou plusieurs tâches de la chaîne. Chaque nom d'attribut d'étiquette fournit les annotations d'une tâche individuelle. Vous pouvez spécifier jusqu'à 5 de ces attributs pour les fichiers manifestes augmentés issus de tâches d'étiquetage en chaîne.

Pour plus d'informations sur les tâches d'étiquetage en chaîne et pour obtenir des exemples des résultats qu'elles produisent, consultez la section Chaining Labeling Jobs dans le manuel Amazon SageMaker AI Developer Guide.

Mode multi-classes

En mode multiclasse, la classification attribue une classe à chaque document. Les cours individuels s'excluent mutuellement. Par exemple, vous pouvez classer un film dans la catégorie comédie ou science-fiction, mais pas dans les deux catégories.

> **Note**
>
> La console Amazon Comprehend fait référence au mode multi-classes en tant que mode à étiquette unique.

Rubriques

- Modèles en texte brut
- Modèles de documents natifs

Modèles en texte brut

Pour entraîner un modèle en texte brut, vous pouvez fournir des données d'entraînement étiquetées sous forme de fichier CSV ou de fichier manifeste augmenté à partir d' SageMaker AI Ground Truth.

Fichier CSV

Pour obtenir des informations générales sur l'utilisation de fichiers CSV pour les classificateurs d'entraînement, consultezfichiers CSV.

Fournissez les données d'entraînement sous forme de fichier CSV à deux colonnes. Pour chaque ligne, la première colonne contient la valeur de l'étiquette de classe. La deuxième colonne contient un exemple de document texte pour cette classe. Chaque ligne doit se terminer par \ n ou \ r \ncaractères.

L'exemple suivant montre un fichier CSV contenant trois documents.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS,Text of document 3
```

L'exemple suivant montre une ligne d'un fichier CSV qui entraîne un classificateur personnalisé pour détecter si un e-mail est du spam :

```
SPAM,"Paulo, your $1000 award is waiting for you! Claim it while you still can at
 http://example.com."
```

Fichier manifeste augmenté

Pour obtenir des informations générales sur l'utilisation de fichiers manifestes augmentés pour l'entraînement des classificateurs, consultezFichier manifeste augmenté.

Pour les documents en texte brut, chaque ligne du fichier manifeste augmenté est un objet JSON complet qui contient un document de formation, un nom de classe unique et d'autres métadonnées provenant de Ground Truth. L'exemple suivant est un fichier manifeste augmenté destiné à entraîner un classificateur personnalisé à reconnaître les spams :

```
{"source":"Document 1 text", "MultiClassJob":0, "MultiClassJob-metadata":
{"confidence":0.62, "job-name":"labeling-job/multiclassjob", "class-name":"not_spam",
 "human-annotated":"yes", "creation-date":"2020-05-21T17:36:45.814354",
 "type":"groundtruth/text-classification"}}
```

```
{"source":"Document 2 text", "MultiClassJob":1, "MultiClassJob-metadata":
{"confidence":0.81, "job-name":"labeling-job/multiclassjob", "class-name":"spam",
 "human-annotated":"yes", "creation-date":"2020-05-21T17:37:51.970530",
 "type":"groundtruth/text-classification"}}
{"source":"Document 3 text", "MultiClassJob":1, "MultiClassJob-metadata":
{"confidence":0.81, "job-name":"labeling-job/multiclassjob", "class-name":"spam",
 "human-annotated":"yes", "creation-date":"2020-05-21T17:37:51.970566",
 "type":"groundtruth/text-classification"}}
```

L'exemple suivant montre un objet JSON issu du fichier manifeste augmenté, formaté dans un souci de lisibilité :

```
{
   "source": "Paulo, your $1000 award is waiting for you! Claim it while you still can
 at http://example.com.",
   "MultiClassJob": 0,
   "MultiClassJob-metadata": {
       "confidence": 0.98,
       "job-name": "labeling-job/multiclassjob",
       "class-name": "spam",
       "human-annotated": "yes",
       "creation-date": "2020-05-21T17:36:45.814354",
       "type": "groundtruth/text-classification"
   }
}
```

Dans cet exemple, l'`source`attribut fournit le texte du document de formation, et l'`MultiClassJob`attribut attribue l'index d'une classe à partir d'une liste de classification. L'`job-name`attribut est le nom que vous avez défini pour la tâche d'étiquetage dans Ground Truth.

Lorsque vous démarrez la tâche de formation au classificateur dans Amazon Comprehend, vous spécifiez le même nom de tâche d'étiquetage.

Modèles de documents natifs

Un modèle de document natif est un modèle que vous entraînez avec des documents natifs (tels que PDF, DOCX et images). Vous fournissez les données d'entraînement sous forme de fichier CSV.

Fichier CSV

Pour obtenir des informations générales sur l'utilisation de fichiers CSV pour les classificateurs d'entraînement, consultezfichiers CSV.

Fournissez les données d'entraînement sous forme de fichier CSV à trois colonnes. Pour chaque ligne, la première colonne contient la valeur de l'étiquette de classe. La deuxième colonne contient le nom de fichier d'un exemple de document pour cette classe. La troisième colonne contient le numéro de page. Le numéro de page est facultatif si le document d'exemple est une image.

L'exemple suivant montre un fichier CSV qui fait référence à trois documents d'entrée.

```
CLASS,input-doc-1.pdf,3
CLASS,input-doc-2.docx,1
CLASS,input-doc-3.png
```

L'exemple suivant montre une ligne d'un fichier CSV qui entraîne un classificateur personnalisé pour détecter si un e-mail est du spam. La page 2 du fichier PDF contient l'exemple de spam.

```
SPAM,email-content-3.pdf,2
```

Mode multi-étiquettes

En mode multi-étiquettes, les classes individuelles représentent différentes catégories qui ne s'excluent pas mutuellement. La classification à étiquettes multiples attribue une ou plusieurs classes à chaque document. Par exemple, vous pouvez classer un film en tant que documentaire et un autre en tant que film de science-fiction, d'action et de comédie.

Pour la formation, le mode multi-étiquettes prend en charge jusqu'à 1 million d'exemples contenant jusqu'à 100 classes uniques.

Rubriques

- Modèles en texte brut
- Modèles de documents natifs

Modèles en texte brut

Pour entraîner un modèle en texte brut, vous pouvez fournir des données d'entraînement étiquetées sous forme de fichier CSV ou de fichier manifeste augmenté à partir d' SageMaker AI Ground Truth.

Fichier CSV

Pour obtenir des informations générales sur l'utilisation de fichiers CSV pour les classificateurs d'entraînement, consultezfichiers CSV.

Fournissez les données d'entraînement sous forme de fichier CSV à deux colonnes. Pour chaque ligne, la première colonne contient les valeurs des étiquettes de classe, et la seconde contient un exemple de document texte pour ces classes. Pour saisir plusieurs classes dans la première colonne, utilisez un séparateur (tel qu'un |) entre chaque classe.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS|CLASS|CLASS,Text of document 3
```

L'exemple suivant montre une ligne d'un fichier CSV qui entraîne un classificateur personnalisé pour détecter les genres dans les résumés de films :

```
COMEDY|MYSTERY|SCIENCE_FICTION|TEEN,"A band of misfit teens become unlikely detectives
 when they discover troubling clues about their high school English teacher. Could the
 strange Mrs. Doe be an alien from outer space?"
```

Le séparateur par défaut entre les noms de classes est un tube (|). Vous pouvez toutefois utiliser un caractère différent comme séparateur. Le délimiteur doit être distinct de tous les caractères des noms de classe. Par exemple, si vos classes sont CLASS_1, CLASS_2 et CLASS_3, le trait de soulignement (_) fait partie du nom de la classe. N'utilisez donc pas de trait de soulignement comme séparateur pour séparer les noms de classe.

Fichier manifeste augmenté

Pour obtenir des informations générales sur l'utilisation de fichiers manifestes augmentés pour l'entraînement des classificateurs, consultezFichier manifeste augmenté.

Pour les documents en texte brut, chaque ligne du fichier manifeste augmenté est un objet JSON complet. Il contient un document de formation, des noms de classes et d'autres métadonnées de Ground Truth. L'exemple suivant est un fichier manifeste augmenté permettant d'entraîner un classificateur personnalisé à détecter les genres dans les résumés de films :

```
{"source":"Document 1 text", "MultiLabelJob":[0,4], "MultiLabelJob-metadata":{"job-
name":"labeling-job/multilabeljob", "class-map":{"0":"action", "4":"drama"}, "human-
annotated":"yes", "creation-date":"2020-05-21T19:02:21.521882", "confidence-map":
{"0":0.66}, "type":"groundtruth/text-classification-multilabel"}}
{"source":"Document 2 text", "MultiLabelJob":[3,6], "MultiLabelJob-metadata":{"job-
name":"labeling-job/multilabeljob", "class-map":{"3":"comedy", "6":"horror"}, "human-
annotated":"yes", "creation-date":"2020-05-21T19:00:01.291202", "confidence-map":
{"1":0.61,"0":0.61}, "type":"groundtruth/text-classification-multilabel"}}
```

```
{"source":"Document 3 text", "MultiLabelJob":[1], "MultiLabelJob-metadata":
{"job-name":"labeling-job/multilabeljob", "class-map":{"1":"action"}, "human-
annotated":"yes", "creation-date":"2020-05-21T18:58:51.662050", "confidence-map":
{"1":0.68}, "type":"groundtruth/text-classification-multilabel"}}
```

L'exemple suivant montre un objet JSON issu du fichier manifeste augmenté, formaté dans un souci de lisibilité :

```
{
      "source": "A band of misfit teens become unlikely detectives when
                 they discover troubling clues about their high school English
 teacher.
                   Could the strange Mrs. Doe be an alien from outer space?",
      "MultiLabelJob": [
          3,
          8,
          10,
          11
      ],
      "MultiLabelJob-metadata": {
          "job-name": "labeling-job/multilabeljob",
          "class-map": {
              "3": "comedy",
              "8": "mystery",
              "10": "science_fiction",
              "11": "teen"
          },
          "human-annotated": "yes",
          "creation-date": "2020-05-21T19:00:01.291202",
          "confidence-map": {
              "3": 0.95,
              "8": 0.77,
              "10": 0.83,
              "11": 0.92
          },
          "type": "groundtruth/text-classification-multilabel"
      }
  }
```

Dans cet exemple, l'`source`attribut fournit le texte du document de formation et `MultiLabelJob` attribue les index de plusieurs classes à partir d'une liste de classification. Le nom de tâche indiqué

dans les `MultiLabelJob` métadonnées est le nom que vous avez défini pour le travail d'étiquetage dans Ground Truth.

Modèles de documents natifs

Un modèle de document natif est un modèle que vous entraînez avec des documents natifs (tels que des fichiers PDF, DOCX et images). Vous fournissez des données d'entraînement étiquetées sous forme de fichier CSV.

Fichier CSV

Pour obtenir des informations générales sur l'utilisation de fichiers CSV pour les classificateurs d'entraînement, consultezfichiers CSV.

Fournissez les données d'entraînement sous forme de fichier CSV à trois colonnes. Pour chaque ligne, la première colonne contient les valeurs des étiquettes de classe. La deuxième colonne contient le nom de fichier d'un exemple de document pour ces classes. La troisième colonne contient le numéro de page. Le numéro de page est facultatif si le document d'exemple est une image.

Pour saisir plusieurs classes dans la première colonne, utilisez un séparateur (tel qu'un |) entre chaque classe.

```
CLASS,input-doc-1.pdf,3
CLASS,input-doc-2.docx,1
CLASS|CLASS|CLASS,input-doc-3.png,2
```

L'exemple suivant montre une ligne d'un fichier CSV qui entraîne un classificateur personnalisé pour détecter les genres dans les résumés de films. La page 2 du fichier PDF contient un exemple de comedy/teen film.

```
COMEDY|TEEN,movie-summary-1.pdf,2
```

Le séparateur par défaut entre les noms de classes est un tube (|). Vous pouvez toutefois utiliser un caractère différent comme séparateur. Le délimiteur doit être distinct de tous les caractères des noms de classe. Par exemple, si vos classes sont CLASS_1, CLASS_2 et CLASS_3, le trait de soulignement (_) fait partie du nom de la classe. N'utilisez donc pas de trait de soulignement comme séparateur pour séparer les noms de classe.

Modèles de classification des formations

Pour entraîner un modèle en vue d'une classification personnalisée, vous définissez les catégories et vous fournissez des exemples de documents pour entraîner le modèle personnalisé. Vous entraînez le modèle en mode multiclasse ou multilabel. Le mode multi-classes associe une seule classe à chaque document. Le mode multi-étiquettes associe une ou plusieurs classes à chaque document.

La classification personnalisée prend en charge deux types de modèles de classificateur : les modèles en texte brut et les modèles de documents natifs. Un modèle en texte brut classe les documents en fonction de leur contenu textuel. Un modèle de document natif classe également les documents en fonction du contenu du texte. Un modèle de document natif peut également utiliser des signaux supplémentaires, tels que ceux issus de la mise en page du document. Vous entraînez un modèle de document natif avec des documents natifs pour que le modèle apprenne les informations de mise en page.

Les modèles en texte brut présentent les caractéristiques suivantes :

- Vous entraînez le modèle à l'aide de documents texte codés en UTF-8.
- Vous pouvez entraîner le modèle à l'aide de documents rédigés dans l'une des langues suivantes : anglais, espagnol, allemand, italien, français ou portugais.
- Les documents de formation relatifs à un classificateur donné doivent tous utiliser le même langage.
- Les documents de formation sont en texte brut, il n'y a donc pas de frais supplémentaires pour l'extraction de texte.

Les modèles de documents natifs présentent les caractéristiques suivantes :

- Vous entraînez le modèle à l'aide de documents semi-structurés, notamment les types de documents suivants :
  - Documents PDF numérisés et numérisés.
  - Documents Word (DOCX).
  - Images : fichiers JPG, fichiers PNG et fichiers TIFF d'une seule page.
  - Fichiers JSON de sortie de l'API Textract.
- Vous entraînez le modèle à l'aide de documents en anglais.

- Si vos documents de formation incluent des fichiers numérisés, des frais supplémentaires vous seront facturés pour l'extraction de texte. Consultez la page de tarification d'Amazon Comprehend pour plus de détails.

Vous pouvez classer tous les types de documents pris en charge en utilisant l'un ou l'autre type de modèle. Toutefois, pour des résultats plus précis, nous recommandons d'utiliser un modèle en texte brut pour classer les documents en texte brut et un modèle de document natif pour classer les documents semi-structurés.

Rubriques

- Entraînez des classificateurs personnalisés (console)
- Entraînez des classificateurs personnalisés (API)
- Testez les données d'entraînement
- Résultat d'entraînement du classificateur
- Métriques de classification personnalisées

Entraînez des classificateurs personnalisés (console)

Vous pouvez créer et entraîner un classificateur personnalisé à l'aide de la console, puis utiliser le classificateur personnalisé pour analyser vos documents.

Pour former un classificateur personnalisé, vous avez besoin d'un ensemble de documents de formation. Vous étiquetez ces documents avec les catégories que vous souhaitez que le classificateur de documents reconnaisse. Pour plus d'informations sur la préparation de vos documents de formation, consultezPréparation des données d'entraînement du classificateur.

Pour créer et entraîner un modèle de classificateur de documents

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Personnalisation, puis Classification personnalisée.
3. Choisissez Créer un nouveau modèle.
4. Sous Paramètres du modèle, entrez le nom du modèle pour le classificateur. Le nom doit être unique au sein de votre compte et de votre région actuelle.

(Facultatif) Entrez un nom de version. Le nom doit être unique au sein de votre compte et de votre région actuelle.

5. Sélectionnez la langue des documents de formation. Pour connaître les langues prises en charge par les classificateurs, consultezModèles de classification des formations.
6. (Facultatif) Si vous souhaitez chiffrer les données du volume de stockage pendant qu'Amazon Comprehend traite votre tâche de formation, choisissez le chiffrement Classifier. Choisissez ensuite d'utiliser une clé KMS associée à votre compte actuel ou une clé provenant d'un autre compte.

   - Si vous utilisez une clé associée au compte actuel, choisissez l'ID de clé pour l'ID de clé KMS.
   - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'ID de clé sous ARN de la clé KMS.

   > **Note**
   >
   > Pour plus d'informations sur la création et l'utilisation des clés KMS et le chiffrement associé, consultez AWS Key Management Service (AWS KMS).

7. Sous Spécifications des données, choisissez le type de modèle d'entraînement à utiliser.

   - Documents en texte brut : choisissez cette option pour créer un modèle en texte brut. Entraînez le modèle à l'aide de documents en texte brut.
   - Documents natifs : choisissez cette option pour créer un modèle de document natif. Entraînez le modèle à l'aide de documents natifs (PDF, Word, images).

8. Choisissez le format de données de vos données d'entraînement. Pour plus d'informations sur les formats de données, consultezFormats de fichiers d'entraînement du classificateur.

   - Fichier CSV : sélectionnez cette option si vos données d'entraînement utilisent le format de fichier CSV.
   - Manifeste augmenté : choisissez cette option si vous avez utilisé Ground Truth pour créer des fichiers de manifeste augmenté pour vos données d'entraînement. Ce format est disponible si vous avez choisi les documents en texte brut comme type de modèle de formation.

9. Choisissez le mode Classificateur à utiliser.

   - Mode étiquette unique : choisissez ce mode si les catégories que vous attribuez aux documents s'excluent mutuellement et que vous entraînez votre classificateur à attribuer une

étiquette à chaque document. Dans l'API Amazon Comprehend, le mode à étiquette unique est appelé mode multi-classes.

- Mode multi-étiquettes : choisissez ce mode si plusieurs catégories peuvent être appliquées simultanément à un document et que vous entraînez votre classificateur à attribuer une ou plusieurs étiquettes à chaque document.

10. Si vous choisissez le mode multi-étiquettes, vous pouvez sélectionner le séparateur pour les étiquettes. Utilisez ce caractère délimiteur pour séparer les libellés lorsqu'un document de formation comporte plusieurs classes. Le séparateur par défaut est le caractère en forme de tube.
11. (Facultatif) Si vous avez choisi le manifeste augmenté comme format de données, vous pouvez saisir jusqu'à cinq fichiers de manifeste augmenté. Chaque fichier manifeste augmenté contient soit un ensemble de données d'entraînement, soit un ensemble de données de test. Vous devez fournir au moins un ensemble de données d'entraînement. Les ensembles de données de test sont facultatifs. Procédez comme suit pour configurer les fichiers manifestes augmentés :

    a. Sous Entraînement et jeu de données de test, développez le panneau de localisation des entrées.
    b. Dans Type d'ensemble de données, choisissez Données d'entraînement ou Données de test.
    c. Pour l'emplacement S3 du fichier manifeste augmenté SageMaker AI Ground Truth, entrez l'emplacement du compartiment Amazon S3 qui contient le fichier manifeste ou naviguez jusqu'à celui-ci en choisissant Browse S3. Le rôle IAM que vous utilisez pour les autorisations d'accès pour le travail de formation doit disposer d'autorisations de lecture pour le compartiment S3.
    d. Pour les noms d'attributs, entrez le nom de l'attribut qui contient vos annotations. Si le fichier contient des annotations provenant de plusieurs tâches d'étiquetage en chaîne, ajoutez un attribut pour chaque tâche.
    e. Pour ajouter un autre emplacement d'entrée, choisissez Ajouter un emplacement d'entrée, puis configurez l'emplacement suivant.

12. (Facultatif) Si vous avez choisi le fichier CSV comme format de données, procédez comme suit pour configurer le jeu de données d'entraînement et le jeu de données de test facultatif :

    a. Sous Ensemble de données d'entraînement, entrez l'emplacement du compartiment Amazon S3 qui contient votre fichier CSV de données d'entraînement ou accédez à celui-ci en choisissant Browse S3. Le rôle IAM que vous utilisez pour les autorisations d'accès pour le travail de formation doit disposer d'autorisations de lecture pour le compartiment S3.

(Facultatif) Si vous avez choisi les documents natifs comme type de modèle de formation, vous fournissez également l'URL du dossier Amazon S3 qui contient les fichiers d'exemple de formation.

b. Sous Test dataset, indiquez si vous souhaitez fournir des données supplémentaires à Amazon Comprehend afin de tester le modèle entraîné.

- Autosplit : Autosplit sélectionne automatiquement 10 % de vos données d'entraînement à réserver pour les utiliser comme données de test.
- (Facultatif) Fourni par le client : entrez l'URL du fichier CSV des données de test dans Amazon S3. Vous pouvez également accéder à son emplacement dans Amazon S3 et choisir Sélectionner un dossier.

  (Facultatif) Si vous avez choisi les documents natifs comme type de modèle de formation, vous devez également fournir l'URL du dossier Amazon S3 qui contient les fichiers de test.

13. (Facultatif) En mode lecture de document, vous pouvez remplacer les actions d'extraction de texte par défaut. Cette option n'est pas requise pour les modèles de texte brut, car elle s'applique à l'extraction de texte pour les documents numérisés. Pour de plus amples informations, veuillez consulter Configuration des options d'extraction de texte.
14. (Facultatif pour les modèles en texte brut) Pour les données de sortie, entrez l'emplacement d'un compartiment Amazon S3 pour enregistrer les données de sortie d'entraînement, telles que la matrice de confusion. Pour de plus amples informations, veuillez consulter Matrice Confusion.

    (Facultatif) Si vous choisissez de chiffrer le résultat de votre formation, choisissez Chiffrement. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

    - Si vous utilisez une clé associée au compte actuel, choisissez l'alias de clé pour l'ID de clé KMS.
    - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'alias ou de l'ID de clé sous ID de clé KMS.
15. Pour le rôle IAM, choisissez Choisir un rôle IAM existant, puis choisissez un rôle IAM existant doté d'autorisations de lecture pour le compartiment S3 contenant vos documents de formation. Le rôle doit avoir une politique de confiance qui commence par « `comprehend.amazonaws.com` pour être valide ».

Si vous ne possédez pas encore de rôle IAM doté de ces autorisations, choisissez Créer un rôle IAM pour en créer un. Choisissez les autorisations d'accès pour accorder ce rôle, puis choisissez un suffixe de nom pour distinguer le rôle des rôles IAM dans votre compte.

> **Note**
>
> Pour les documents d'entrée chiffrés, le rôle IAM utilisé doit également disposer d'une `kms:Decrypt` autorisation. Pour de plus amples informations, veuillez consulter Autorisations requises pour utiliser le chiffrement KMS.

16. (Facultatif) Pour lancer vos ressources dans Amazon Comprehend à partir d'un VPC, entrez l'ID du VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseaux (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.
    2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

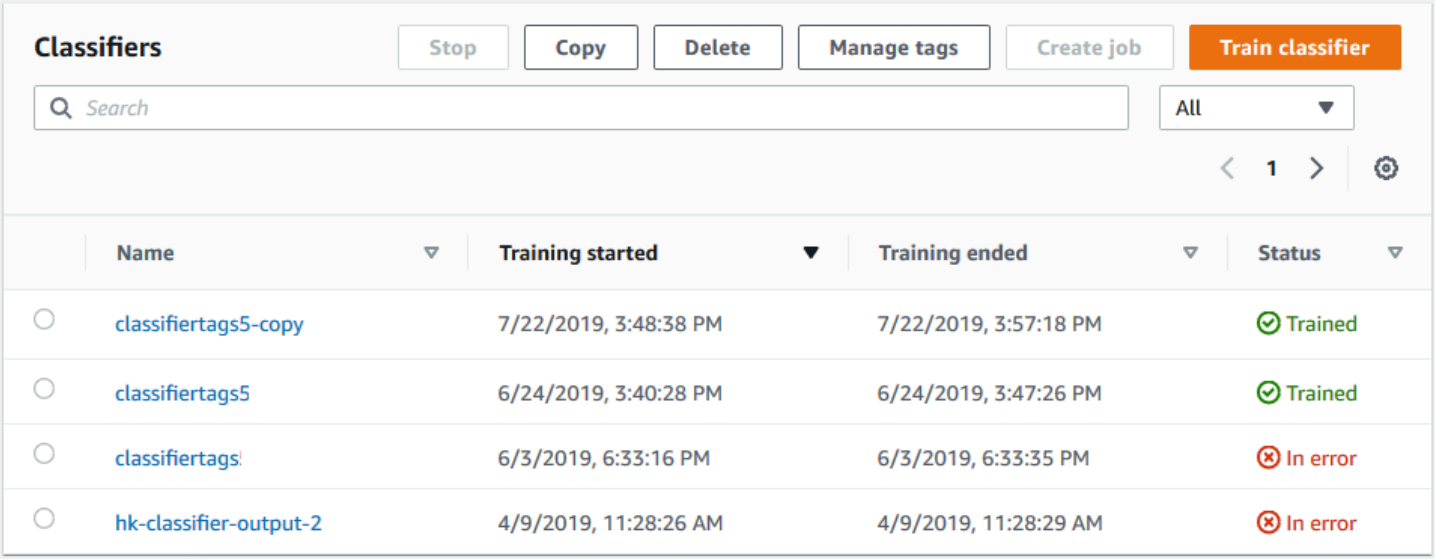
    > **Note**
    >
    > Lorsque vous utilisez un VPC pour votre tâche de classification, le VPC `DataAccessRole` utilisé pour les opérations de création et de démarrage doit disposer d'autorisations sur le VPC qui accède aux documents d'entrée et au compartiment de sortie.

17. (Facultatif) Pour ajouter une balise au classificateur personnalisé, entrez une paire clé-valeur sous Tags. Choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le classificateur, choisissez Supprimer le tag. Pour de plus amples informations, veuillez consulter Balisage de vos ressources .
18. Choisissez Créer.

La console affiche la page Classificateurs. Le nouveau classificateur apparaît dans le tableau et indique son `Submitted` statut. Lorsque le classificateur commence à traiter les documents de formation, le statut passe à`Training`. Lorsqu'un classificateur est prêt à être utilisé, son statut

passe à `Trained` ou`Trained with warnings`. Si le statut est le cas`TRAINED_WITH_WARNINGS`, passez en revue le dossier des fichiers ignorés dans leRésultat d'entraînement du classificateur.

Si Amazon Comprehend a rencontré des erreurs lors de la création ou de la formation, le statut passe à. `In error` Vous pouvez choisir une tâche de classificateur dans le tableau pour obtenir plus d'informations sur le classificateur, y compris les éventuels messages d'erreur.



Entraînez des classificateurs personnalisés (API)

Pour créer et entraîner un classificateur personnalisé, utilisez l'CreateDocumentClassifieropération.

Vous pouvez suivre la progression de la demande à l'aide de l'DescribeDocumentClassifieropération. Une fois le `Status` champ passé à`TRAINED`, vous pouvez utiliser le classificateur pour classer les documents. Si le statut est le cas`TRAINED_WITH_WARNINGS`, passez en revue le dossier des fichiers ignorés dans le formulaire Résultat d'entraînement du classificateur de l'`CreateDocumentClassifier`opération.

Rubriques

- Classification personnalisée de la formation à l'aide du AWS Command Line Interface
- Utilisation du SDK AWS SDK pour Java or pour Python

Classification personnalisée de la formation à l'aide du AWS Command Line Interface

Les exemples suivants montrent comment utiliser l'`CreateDocumentClassifier`opération, l'`DescribeDocumentClassificationJob`opération et tout autre classificateur personnalisé APIs avec le AWS CLI.

Les exemples sont formatés pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Créez un classificateur personnalisé en texte brut à l'aide de l'opération. `create-document-classifier`

```
aws comprehend create-document-classifier \
     --region region \
     --document-classifier-name testDelete \
     --language-code en \
     --input-data-config S3Uri=s3://S3Bucket/docclass/file name \
     --data-access-role-arn arn:aws:iam::account number:role/testFlywheelDataAccess
```

Pour créer un classificateur personnalisé natif, fournissez les paramètres supplémentaires suivants dans la `create-document-classifier` demande.

1. DocumentType: définissez la valeur sur SEMI_STRUCTURED_DOCUMENT.
2. Documents : emplacement S3 pour les documents de formation (et, éventuellement, les documents de test).
3. OutputDataConfig: indiquez l'emplacement S3 des documents de sortie (et une clé KMS optionnelle).
4. DocumentReaderConfig: champ facultatif pour les paramètres d'extraction de texte.

```
aws comprehend create-document-classifier \
     --region region \
     --document-classifier-name testDelete \
     --language-code en \
     --input-data-config
          S3Uri=s3://S3Bucket/docclass/file name \
           DocumentType \
             Documents  \
     --output-data-config S3Uri=s3://S3Bucket/docclass/file name \
     --data-access-role-arn arn:aws:iam::account number:role/testFlywheelDataAccess
```

Obtenez des informations sur un classificateur personnalisé avec l'ARN du classificateur de documents à l'aide de l'`DescribeDocumentClassifier`opération.

```
aws comprehend describe-document-classifier \
```

```
     --region region \
     --document-classifier-arn arn:aws:comprehend:region:account number:document-
classifier/file name
```

Supprimez un classificateur personnalisé à l'aide de cette `DeleteDocumentClassifier` opération.

```
aws comprehend delete-document-classifier \
     --region region \
     --document-classifier-arn arn:aws:comprehend:region:account number:document-
classifier/testDelete
```

Répertoriez tous les classificateurs personnalisés du compte à l'aide de l'`ListDocumentClassifiers`opération.

```
aws comprehend list-document-classifiers
     --region region
```

Utilisation du SDK AWS SDK pour Java or pour Python

Pour des exemples de SDK expliquant comment créer et entraîner un classificateur personnalisé, consultez. Utilisation `CreateDocumentClassifier` avec un AWS SDK ou une CLI

Testez les données d'entraînement

Après avoir entraîné le modèle, Amazon Comprehend teste le modèle de classificateur personnalisé. Si vous ne fournissez pas de jeu de données de test, Amazon Comprehend entraîne le modèle avec 90 % des données de formation. Il réserve 10 % des données de formation à des fins de test. Si vous fournissez un ensemble de données de test, les données de test doivent inclure au moins un exemple pour chaque étiquette unique du jeu de données d'entraînement.

Le test du modèle vous fournit des mesures que vous pouvez utiliser pour estimer la précision du modèle. La console affiche les métriques dans la section Performances du classificateur de la page de détails du classificateur de la console. Ils sont également renvoyés dans les `Metrics` champs renvoyés par l'DescribeDocumentClassifieropération.

Dans les exemples de données de formation suivants, il existe cinq labels : DOCUMENTARY, DOCUMENTARY, SCIENCE_FICTION, DOCUMENTARY, ROMANTIC_COMEDY. Il existe trois catégories uniques : DOCUMENTAIRE, SCIENCE_FICTION, COMÉDIE ROMANTIQUE.

Colonne 1	Colonne 2
DOCUMENTAIRE	texte du document 1
DOCUMENTAIRE	texte du document 2
SCIENCE-FICTION	texte du document 3
DOCUMENTAIRE	texte du document 4
COMÉDIE ROMANTIQUE	texte du document 5

Pour le fractionnement automatique (où Amazon Comprehend réserve 10 % des données de formation à utiliser pour les tests), si les données d'entraînement contiennent un nombre limité d'exemples d'une étiquette spécifique, le jeu de données de test peut ne contenir aucun exemple de cette étiquette. Par exemple, si le jeu de données d'entraînement contient 1 000 instances de la classe DOCUMENTARY, 900 instances de SCIENCE_FICTION et une seule instance de la classe ROMANTIC_COMEDY, le jeu de données de test peut contenir 100 instances DOCUMENTARY et 90 instances SCIENCE_FICTION, mais aucune instance ROMANTIC_COMEDY, car un seul exemple est disponible.

Une fois que vous avez terminé l'entraînement de votre modèle, les métriques d'entraînement fournissent des informations que vous pouvez utiliser pour déterminer si le modèle est suffisamment précis pour vos besoins.

Résultat d'entraînement du classificateur

Une fois qu'Amazon Comprehend a terminé l'apprentissage du modèle de classificateur personnalisé, il crée des fichiers de sortie dans l'emplacement de sortie Amazon S3 que vous avez spécifié dans la demande d'CreateDocumentClassifierAPI ou dans la demande de console équivalente.

Amazon Comprehend crée une matrice de confusion lorsque vous entraînez un modèle en texte brut ou un modèle de document natif. Il peut créer des fichiers de sortie supplémentaires lorsque vous entraînez un modèle de document natif.

Rubriques

- Matrice Confusion
- Sorties supplémentaires pour les modèles de documents natifs

Matrice Confusion

Lorsque vous entraînez un modèle de classificateur personnalisé, Amazon Comprehend crée une matrice de confusion qui fournit des indicateurs sur les performances du modèle pendant l'entraînement. Cette matrice montre une matrice d'étiquettes prédites par le modèle, par rapport aux étiquettes réelles du document. Amazon Comprehend utilise une partie des données de formation pour créer la matrice de confusion.

Une matrice de confusion indique quelles classes pourraient utiliser davantage de données pour améliorer les performances du modèle. Une classe présentant une fraction élevée de prédictions correctes a le plus grand nombre de résultats le long de la diagonale de la matrice. Si le nombre sur la diagonale est inférieur, la classe a une fraction plus faible de prédictions correctes. Vous pouvez ajouter d'autres exemples d'entraînement pour cette classe et entraîner à nouveau le modèle. Par exemple, si 40 % des échantillons de l'étiquette A sont classés dans l'étiquette D, l'ajout d'échantillons supplémentaires pour l'étiquette A et l'étiquette D améliore les performances du classificateur.

Une fois qu'Amazon Comprehend a créé le modèle de classificateur, la matrice de confusion est disponible dans le `confusion_matrix.json` fichier à l'emplacement de sortie S3.

Le format de la matrice de confusion varie selon que vous avez entraîné votre classificateur à utiliser le mode multi-classes ou le mode multi-étiquettes.

Rubriques

- Matrice de confusion pour le mode multiclasse
- Matrice de confusion pour le mode multi-étiquettes

Matrice de confusion pour le mode multiclasse

En mode multiclasse, les classes individuelles s'excluent mutuellement, de sorte que la classification attribue une étiquette à chaque document. Par exemple, un animal peut être un chien ou un chat, mais pas les deux à la fois.

Prenons l'exemple suivant de matrice de confusion pour un classificateur entraîné à plusieurs classes :

```
  A B X Y <-(predicted label)
A 1 2 0 4
B 0 3 0 1
```

```
X 0 0 1 0
Y 1 1 1 1
^
|
(actual label)
```

Dans ce cas, le modèle a prédit ce qui suit :

- Une étiquette « A » a été prédite avec précision, deux étiquettes « A » ont été incorrectement prédites comme des étiquettes « B » et quatre étiquettes « A » ont été incorrectement prédites comme des étiquettes « Y ».
- Trois étiquettes « B » ont été prédites avec précision, et une étiquette « B » a été incorrectement prédite en tant qu'étiquette « Y ».
- Un « X » a été prédit avec précision.
- Une étiquette « Y » a été prédite avec précision, une a été incorrectement prédite comme une étiquette « A », une a été incorrectement prédite comme une étiquette « B » et une autre a été incorrectement prédite comme une étiquette « X ».

La ligne diagonale de la matrice (A:A, B:B, X:X et Y:Y) montre les prédictions précises. Les erreurs de prédiction sont les valeurs situées en dehors de la diagonale. Dans ce cas, la matrice indique les taux d'erreur de prédiction suivants :

- Étiquettes A : 86 %
- Étiquettes B : 25 %
- Étiquettes X : 0 %
- Étiquettes Y : 75 %

Le classificateur renvoie la matrice de confusion sous forme de fichier au format JSON. Le fichier JSON suivant représente la matrice de l'exemple précédent.

```
{
 "type": "multi_class",
 "confusion_matrix": [
 [1, 2, 0,4],
 [0, 3, 0, 1],
 [0, 0, 1, 0],
 [1, 1, 1, 1]],
 "labels": ["A", "B", "X", "Y"],
```

```
 "all_labels": ["A", "B", "X", "Y"]
}
```

Matrice de confusion pour le mode multi-étiquettes

En mode multi-étiquettes, la classification peut attribuer une ou plusieurs classes à un document. Prenons l'exemple suivant d'une matrice de confusion pour un classificateur entraîné à plusieurs classes.

Dans cet exemple, trois libellés sont possibles : `ComedyAction`, et`Drama`. La matrice de confusion multi-étiquettes crée une matrice 2x2 pour chaque étiquette.

```
Comedy                   Action                   Drama
     No Yes                   No Yes                   No Yes   <-(predicted label)

 No  2   1                 No  1   1                 No  3   0

Yes  0   2                Yes  2   1                Yes  1   1
 ^                         ^                        ^
 |                         |                        |
 |-----------(was this label actually used)--------|
```

Dans ce cas, le modèle a renvoyé ce qui suit pour l'`Comedy`étiquette :

- Deux cas où la présence d'une `Comedy` étiquette a été prédite avec précision. Vrai positif (TP).
- Deux cas où l'absence d'une `Comedy` étiquette a été prédite avec précision. Vrai négatif (TN).
- Aucune occurrence où la présence d'une `Comedy` étiquette a été prédite par erreur. Faux positif (FP).
- Un cas où l'absence d'une `Comedy` étiquette a été incorrectement prédite. Faux négatif (FN).

Comme dans le cas d'une matrice de confusion à classes multiples, la ligne diagonale de chaque matrice indique les prédictions exactes.

Dans ce cas, le modèle a prédit avec précision `Comedy` les étiquettes 80 % du temps (TP plus TN) et les a incorrectement prédites 20 % du temps (FP plus FN).

Le classificateur renvoie la matrice de confusion sous forme de fichier au format JSON. Le fichier JSON suivant représente la matrice de l'exemple précédent.

```
{
"type": "multi_label",
"confusion_matrix": [
 [[2, 1],
 [0, 2]],
 [[1, 1],
 [2, 1]],
 [[3, 0],
 [1, 1]]
],
"labels": ["Comedy", "Action", "Drama"]
"all_labels": ["Comedy", "Action", "Drama"]
}
```

Sorties supplémentaires pour les modèles de documents natifs

Amazon Comprehend peut créer des fichiers de sortie supplémentaires lorsque vous entraînez un modèle de document natif.

Sortie Amazon Textract

Si Amazon Comprehend a invoqué Amazon APIs Textract pour extraire le texte de l'un des documents de formation, il enregistre les fichiers de sortie Amazon Textract dans l'emplacement de sortie S3. Il utilise la structure de répertoire suivante :

- Documents de formation :

  `amazon-textract-output/train/<file_name>/<page_num>/textract_output.json`
- Documents de test :

  `amazon-textract-output/test/<file_name>/<page_num>/textract_output.json`

Amazon Comprehend remplit le dossier de test si vous avez fourni des documents de test dans la demande d'API.

Défaillances d'annotation de documents

Amazon Comprehend crée les fichiers suivants dans l'emplacement de sortie Amazon S3 (dans le dossier skipped_documents/) en cas d'échec des annotations :

- failed_annotations_train.jsonl

Le fichier existe si des annotations ont échoué dans les données d'entraînement.

- failed_annotations_test.jsonl

  Le fichier existe si la demande incluait des données de test et que les annotations échouaient dans les données de test.

Les fichiers d'annotation ayant échoué sont des fichiers JSONL au format suivant :

```
{
     "File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
    {"File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."
  }
```

Métriques de classification personnalisées

Amazon Comprehend fournit des statistiques pour vous aider à estimer les performances d'un classificateur personnalisé. Amazon Comprehend calcule les métriques à l'aide des données de test issues de la tâche de formation du classificateur. Les métriques représentent avec précision les performances du modèle pendant l'entraînement, de sorte qu'elles se rapprochent des performances du modèle pour la classification de données similaires.

Utilisez des opérations d'API, par exemple DescribeDocumentClassifierpour récupérer les métriques d'un classificateur personnalisé.

> **Note**
>
> Reportez-vous à la section Métriques : précision, rappel et FScore pour comprendre les mesures sous-jacentes de précision, de rappel et de score F1. Ces métriques sont définies au niveau de la classe. Amazon Comprehend utilise la moyenne des macros pour combiner ces indicateurs dans les ensembles de tests P, R et F1, comme indiqué ci-dessous.

Rubriques

- Métriques
- Améliorer les performances de votre classificateur personnalisé

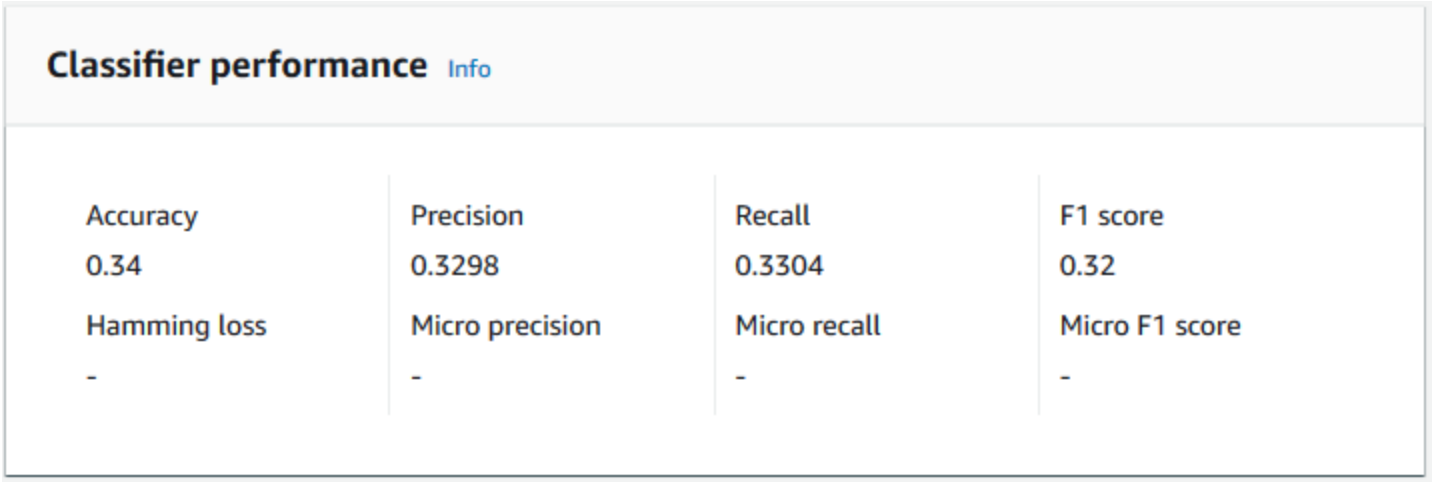
Métriques

Amazon Comprehend prend en charge les métriques suivantes :

Rubriques

- Précision
- Précision (macroprécision)
- Rappel (rappel de macros)
- Score F1 (score F1 macro)
- Défaite de Hamming
- Microprécision
- Micro rappel
- Score de Micro F1

Pour consulter les métriques d'un classificateur, ouvrez la page Détails du classificateur dans la console.



Précision

La précision indique le pourcentage d'étiquettes à partir des données de test que le modèle a prédit avec précision. Pour calculer la précision, divisez le nombre d'étiquettes prédites avec précision dans les documents de test par le nombre total d'étiquettes dans les documents de test.

Par exemple

Étiquette réelle	Étiquette prévue	Exactif/Incorrect
1	1	Précis
0	1	Incorrect
2	3	Incorrect
3	3	Précis
2	2	Précis
1	1	Précis
3	3	Précis

La précision correspond au nombre de prédictions exactes divisé par le nombre total d'échantillons de test = 5/7 = 0,714, soit 71,4 %

Précision (macroprécision)

La précision est une mesure de l'utilité des résultats du classificateur dans les données de test. Il est défini comme le nombre de documents classés avec précision, divisé par le nombre total de classifications pour la classe. La haute précision signifie que le classificateur a renvoyé des résultats nettement plus pertinents que des résultats non pertinents.

La `Precision` métrique est également connue sous le nom de Macro Precision.

L'exemple suivant montre les résultats de précision pour un ensemble de tests.

Étiquette	Taille de l'échantillon	Précision de l'étiquette
Étiquette_1	400	0.75
Étiquette_2	300	0,80
Étiquette_3	30 000	0.90
Étiquette_4	20	0.50

Étiquette	Taille de l'échantillon	Précision de l'étiquette
Étiquette_5	10	0,40

La métrique de précision (précision macro) du modèle est donc la suivante :

```
Macro Precision = (0.75 + 0.80 + 0.90 + 0.50 + 0.40)/5 = 0.67
```

Rappel (rappel de macros)

Cela indique le pourcentage de catégories correctes dans votre texte que le modèle peut prévoir. Cette métrique provient de la moyenne des scores de rappel de toutes les étiquettes disponibles. Le rappel est une mesure de l'exhaustivité des résultats du classificateur pour les données de test.

Un taux de rappel élevé signifie que le classificateur a renvoyé la plupart des résultats pertinents.

La `Recall` métrique est également connue sous le nom de Macro Recall.

L'exemple suivant montre les résultats du rappel d'un ensemble de tests.

Étiquette	Taille de l'échantillon	Rappel d'étiquettes
Étiquette_1	400	0,70
Étiquette_2	300	0,70
Étiquette_3	30 000	0,98
Étiquette_4	20	0,80
Étiquette_5	10	0.10

La métrique de rappel (rappel de macros) du modèle est donc la suivante :

```
Macro Recall = (0.70 + 0.70 + 0.98 + 0.80 + 0.10)/5 = 0.656
```

Score F1 (score F1 macro)

Le score F1 est dérivé des `Recall` valeurs `Precision` et. Il mesure la précision globale du classificateur. Le score le plus élevé est 1 et le score le plus bas est 0.

Amazon Comprehend calcule le score Macro F1. Il s'agit de la moyenne non pondérée des scores F1 du label. En utilisant le kit de test suivant comme exemple :

Étiquette	Taille de l'échantillon	Libellé : score F1
Étiquette_1	400	0,724
Étiquette_2	300	0,824
Étiquette_3	30 000	0,94
Étiquette_4	20	0,62
Étiquette_5	10	0,16

Le score F1 (score Macro F1) du modèle est calculé comme suit :

```
Macro F1 Score = (0.724 + 0.824 + 0.94 + 0.62 + 0.16)/5 = 0.6536
```

Défaite de Hamming

Fraction d'étiquettes incorrectement prédites. Également considéré comme la fraction d'étiquettes incorrectes par rapport au nombre total d'étiquettes. Des scores proches de zéro sont meilleurs.

Microprécision

Original :

Similaire à la métrique de précision, sauf que la microprécision est basée sur le score global de tous les scores de précision additionnés.

Micro rappel

Similaire à la métrique de rappel, sauf que le micro rappel est basé sur le score global de tous les scores de rappel additionnés.

Score de Micro F1

Le score Micro F1 est une combinaison des métriques Micro Precision et Micro Recall.

Améliorer les performances de votre classificateur personnalisé

Les métriques fournissent un aperçu des performances de votre classificateur personnalisé lors d'une tâche de classification. Si les indicateurs sont faibles, le modèle de classification risque de ne pas être efficace pour votre cas d'utilisation. Plusieurs options s'offrent à vous pour améliorer les performances de votre classificateur :

1. Dans vos données d'entraînement, fournissez des exemples concrets qui définissent une séparation claire des catégories. Par exemple, fournissez des documents qui utilisent le caractère unique words/sentences pour représenter la catégorie.
2. Ajoutez des données supplémentaires pour les étiquettes sous-représentées dans vos données d'entraînement.
3. Essayez de réduire le biais dans les catégories. Si la plus grande étiquette de vos données contient plus de 10 fois le nombre de documents figurant dans la plus petite étiquette, essayez d'augmenter le nombre de documents pour la plus petite étiquette. Assurez-vous de réduire le rapport d'asymétrie à 10:1 au maximum entre les classes les plus représentées et les moins représentées. Vous pouvez également essayer de supprimer les documents d'entrée des classes les plus représentées.

Exécution d'analyses en temps réel

Après avoir entraîné un classificateur personnalisé, vous pouvez classer les documents à l'aide d'une analyse en temps réel. L'analyse en temps réel prend un seul document en entrée et renvoie les résultats de manière synchrone. La classification personnalisée accepte divers types de documents comme entrées pour une analyse en temps réel. Pour en savoir plus, consultez Entrées pour une analyse personnalisée en temps réel.

Si vous prévoyez d'analyser des fichiers image ou des documents PDF numérisés, votre politique IAM doit autoriser l'utilisation de deux méthodes d'API Amazon Textract DetectDocumentText (AnalyzeDocumentet). Amazon Comprehend invoque ces méthodes lors de l'extraction de texte. Pour un exemple de politique, consultez Autorisations requises pour effectuer des actions d'analyse de documents.

Vous devez créer un point de terminaison pour exécuter une analyse en temps réel à l'aide d'un modèle de classification personnalisé.

Rubriques

- Analyse en temps réel pour une classification personnalisée (console)
- Analyse en temps réel pour une classification personnalisée (API)
- Sorties pour une analyse en temps réel

Analyse en temps réel pour une classification personnalisée (console)

Vous pouvez utiliser la console Amazon Comprehend pour exécuter une analyse en temps réel à l'aide d'un modèle de classification personnalisé.

Vous créez un point de terminaison pour exécuter l'analyse en temps réel. Un point de terminaison inclut des ressources gérées qui rendent votre modèle personnalisé disponible pour une inférence en temps réel.

Pour plus d'informations sur le provisionnement du débit des terminaux et les coûts associés, consultez. Utilisation des points de terminaison Amazon Comprehend

Rubriques

- Création d'un point de terminaison pour une classification personnalisée
- Exécution d'une classification personnalisée en temps réel

Création d'un point de terminaison pour une classification personnalisée

Pour créer un point de terminaison (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Endpoints, puis cliquez sur le bouton Créer un point de terminaison. L'écran Créer un point de terminaison s'ouvre.
3. Donnez un nom au point de terminaison. Le nom doit être unique dans la région et le compte actuels.
4. Choisissez un modèle personnalisé auquel vous souhaitez associer le nouveau point de terminaison. Dans le menu déroulant, vous pouvez effectuer une recherche par nom de modèle.

> **Note**
>
> Vous devez créer un modèle avant de pouvoir y associer un point de terminaison. Si vous n'avez pas encore de modèle, consultezModèles de classification des formations.

5. (Facultatif) Pour ajouter une balise au point de terminaison, entrez une paire clé-valeur sous Tags et choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le point de terminaison, choisissez Supprimer le tag
6. Entrez le nombre d'unités d'inférence (IUs) à attribuer au point de terminaison. Chaque unité représente un débit de 100 caractères par seconde pour un maximum de deux documents par seconde. Pour plus d'informations sur le débit des terminaux, consultezUtilisation des points de terminaison Amazon Comprehend.
7. (Facultatif) Si vous créez un nouveau point de terminaison, vous avez la possibilité d'utiliser l'estimateur IU. En fonction du débit ou du nombre de caractères que vous souhaitez analyser par seconde, il peut être difficile de savoir de combien d'unités d'inférence vous avez besoin. Cette étape facultative peut vous aider à déterminer le nombre de IUs à demander.
8. À partir du récapitulatif des achats, passez en revue le coût horaire, quotidien et mensuel estimé de votre terminal.
9. Cochez cette case si vous comprenez que votre compte est débité pour le terminal à partir du moment où il démarre et jusqu'à ce que vous le supprimiez.
10. Choisissez Créer un point de terminaison

Exécution d'une classification personnalisée en temps réel

Une fois que vous avez créé un point de terminaison, vous pouvez exécuter une analyse en temps réel à l'aide de votre modèle personnalisé. Il existe deux méthodes pour exécuter une analyse en temps réel à partir de la console. Vous pouvez saisir du texte ou télécharger un fichier, comme indiqué ci-dessous.

Pour exécuter une analyse en temps réel à l'aide d'un modèle personnalisé (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, sélectionnez Analyse en temps réel.
3. Sous Type d'entrée, choisissez Personnalisé pour le type d'analyse.

4. Sous Type de modèle personnalisé, choisissez Classification personnalisée.
5. Pour Endpoint, choisissez le point de terminaison que vous souhaitez utiliser. Ce point de terminaison est lié à un modèle personnalisé spécifique.
6. Pour spécifier les données d'entrée à analyser, vous pouvez saisir du texte ou télécharger un fichier.
   - Pour saisir du texte :
     a. Choisissez Saisie de texte.
     b. Entrez le texte que vous souhaitez analyser.
   - Pour télécharger un fichier, procédez comme suit :
     a. Choisissez Charger le fichier et entrez le nom du fichier à télécharger.
     b. (Facultatif) Sous Actions de lecture avancées, vous pouvez remplacer les actions par défaut pour l'extraction de texte. Pour plus d'informations, consultez Configuration des options d'extraction de texte.

   Pour de meilleurs résultats, faites correspondre le type d'entrée au type de modèle de classificateur. La console affiche un avertissement si vous soumettez un document natif vers un modèle de texte brut ou du texte brut vers un modèle de document natif. Pour de plus amples informations, veuillez consulter Modèles de classification des formations.
7. Choisissez Analyser. Amazon Comprehend analyse les données d'entrée à l'aide de votre modèle personnalisé. Amazon Comprehend affiche les classes découvertes, ainsi qu'une évaluation de confiance pour chaque classe.

Analyse en temps réel pour une classification personnalisée (API)

Vous pouvez utiliser l'API Amazon Comprehend pour exécuter une classification en temps réel avec un modèle personnalisé. Tout d'abord, vous créez un point de terminaison pour exécuter l'analyse en temps réel. Après avoir créé le point de terminaison, vous exécutez la classification en temps réel.

Les exemples présentés dans cette section utilisent des formats de commande pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Pour plus d'informations sur le provisionnement du débit des terminaux et les coûts associés, consultez. Utilisation des points de terminaison Amazon Comprehend

Rubriques

- Création d'un point de terminaison pour une classification personnalisée
- Exécution d'une classification personnalisée en temps réel

Création d'un point de terminaison pour une classification personnalisée

L'exemple suivant montre le fonctionnement de l'CreateEndpointAPI à l'aide du AWS CLI.

```
aws comprehend create-endpoint \
    --desired-inference-units number of inference units \
    --endpoint-name endpoint name \
    --model-arn arn:aws:comprehend:region:account-id:model/example \
    --tags Key=My1stTag,Value=Value1
```

Amazon Comprehend répond comme suit :

```
{
   "EndpointArn": "Arn"
}
```

Exécution d'une classification personnalisée en temps réel

Après avoir créé un point de terminaison pour votre modèle de classification personnalisé, vous l'utilisez pour exécuter l'opération ClassifyDocumentd'API. Vous pouvez saisir du texte à l'aide du `bytes` paramètre `text` or. Entrez les autres types d'entrée à l'aide du `bytes` paramètre.

Pour les fichiers image et les fichiers PDF, vous pouvez utiliser le `DocumentReaderConfig` paramètre pour remplacer les actions d'extraction de texte par défaut. Pour plus d'informations, consultez Configuration des options d'extraction de texte.

Pour de meilleurs résultats, faites correspondre le type d'entrée au type de modèle de classificateur. La réponse de l'API inclut un avertissement si vous soumettez un document natif à un modèle de texte brut ou un fichier de texte brut à un modèle de document natif. Pour de plus amples informations, veuillez consulter Modèles de classification des formations.

À l'aide du AWS Command Line Interface

Les exemples suivants montrent comment utiliser la commande CLI classify-document.

Classez le texte à l'aide du AWS CLI

L'exemple suivant exécute une classification en temps réel sur un bloc de texte.

```
aws comprehend classify-document \
     --endpoint-arn arn:aws:comprehend:region:account-id:endpoint/endpoint name \
     --text 'From the Tuesday, April 16th, 1912 edition of The Guardian newspaper: The
 maiden voyage of the White Star liner Titanic,
     the largest ship ever launched ended in disaster. The Titanic started her trip
 from Southampton for New York on Wednesday. Late
     on Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
 wireless telegraphy she sent out signals of distress,
     and several liners were near enough to catch and respond to the call.'
```

Amazon Comprehend répond comme suit :

```
{
    "Classes": [
       {
          "Name": "string",
          "Score": 0.9793661236763
       }
    ]
 }
```

Classez un document semi-structuré à l'aide du AWS CLI

Pour analyser la classification personnalisée d'un fichier PDF, Word ou image, exécutez la `classify-document` commande avec le fichier d'entrée dans le `bytes` paramètre.

L'exemple suivant utilise une image comme fichier d'entrée. Il utilise l'`file`boption permettant de coder en base 64 les octets du fichier image. Pour plus d'informations, consultez la section Objets binaires volumineux dans le guide de AWS Command Line Interface l'utilisateur.

Cet exemple transmet également un fichier JSON nommé `config.json` pour définir les options d'extraction de texte.

```
$ aws comprehend classify-document \
> --endpoint-arn arn \
> --language-code en \
> --bytes fileb://image1.jpg   \
```

```
> --document-reader-config file://config.json
```

Le fichier config.json contient le contenu suivant.

```
{
   "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
   "DocumentReadAction": "TEXTRACT_DETECT_DOCUMENT_TEXT"
}
```

Amazon Comprehend répond comme suit :

```
{
    "Classes": [
       {
          "Name": "string",
          "Score": 0.9793661236763
       }
    ]
}
```

Pour plus d'informations, consultez le ClassifyDocumentmanuel Amazon Comprehend API Reference.

Sorties pour une analyse en temps réel

Sorties pour entrées de texte

Pour les entrées de texte, la sortie inclut la liste des classes ou des étiquettes identifiées par l'analyse du classificateur. L'exemple suivant montre une liste de deux classes.

```
"Classes": [
  {
     "Name": "abc",
     "Score": 0.2757999897003174,
     "Page": 1
  },
  {
    "Name": "xyz",
```

```
    "Score": 0.2721000015735626,
    "Page": 1
  }
]
```

Sorties pour entrées semi-structurées

Pour un document d'entrée semi-structuré ou un fichier texte, la sortie peut inclure les champs supplémentaires suivants :

- DocumentMetadata — Informations d'extraction relatives au document. Les métadonnées incluent une liste de pages du document, avec le nombre de caractères extraits de chaque page. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- DocumentType — Type de document pour chaque page du document d'entrée. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- Erreurs : erreurs au niveau de la page détectées par le système lors du traitement du document d'entrée. Le champ est vide si le système n'a détecté aucune erreur.
- Avertissements — Avertissements détectés lors du traitement du document d'entrée. La réponse inclut un avertissement en cas de non-concordance entre le type de document d'entrée et le type de modèle associé au point de terminaison que vous avez spécifié. Le champ est vide si le système n'a généré aucun avertissement.

Pour plus de détails sur ces champs de sortie, consultez ClassifyDocumentle manuel Amazon Comprehend API Reference.

L'exemple suivant montre le résultat d'un document d'entrée PDF natif d'une page.

```
{
  "Classes": [
      {
          "Name": "123",
          "Score": 0.39570000767707825,
          "Page": 1
      },
      {
          "Name": "abc",
          "Score": 0.2757999897003174,
          "Page": 1
      },
```

```
        {
            "Name": "xyz",
            "Score": 0.2721000015735626,
            "Page": 1
        }
    ],
    "DocumentMetadata": {
        "Pages": 1,
        "ExtractedCharacters": [
            {
                "Page": 1,
                "Count": 2013
            }
        ]
    },
    "DocumentType": [
        {
            "Page": 1,
            "Type": "NATIVE_PDF"
        }
    ]
}
```

Exécution de tâches asynchrones

Après avoir entraîné un classificateur personnalisé, vous pouvez utiliser des tâches asynchrones pour analyser des documents volumineux ou plusieurs documents en un seul lot.

La classification personnalisée accepte différents types de documents d'entrée. Pour en savoir plus, consultez Entrées pour une analyse personnalisée asynchrone.

Si vous prévoyez d'analyser des fichiers image ou des documents PDF numérisés, votre politique IAM doit autoriser l'utilisation de deux méthodes d'API Amazon Textract DetectDocumentText (AnalyzeDocumentet). Amazon Comprehend invoque ces méthodes lors de l'extraction de texte. Pour un exemple de politique, consultez Autorisations requises pour effectuer des actions d'analyse de documents.

Pour la classification de documents semi-structurés (fichiers image, PDF ou Docx) à l'aide d'un modèle en texte brut, utilisez le format d'entrée. `one document per file` Incluez également le `DocumentReaderConfig` paramètre dans votre StartDocumentClassificationJobdemande.

Rubriques

- Formats de fichiers pour l'analyse asynchrone
- Tâches d'analyse pour une classification personnalisée (console)
- Tâches d'analyse pour une classification personnalisée (API)
- Sorties pour les tâches d'analyse asynchrones

Formats de fichiers pour l'analyse asynchrone

Lorsque vous exécutez une analyse asynchrone avec votre modèle, vous avez le choix entre plusieurs formats pour les documents d'entrée : `One document per line` ou`one document per file`. Le format que vous utilisez dépend du type de document que vous souhaitez analyser, comme décrit dans le tableau suivant.

Description	Format
L'entrée contient plusieurs fichiers. Chaque fichier contient un document d'entrée. Ce format est idéal pour les collections de documents volumineux, tels que des articles de journaux ou des articles scientifiques. Utilisez également ce format pour les documents semi-structurés (fichiers image, PDF ou Docx) à l'aide d'un classificateur de documents natif.	Un document par fichier
L'entrée est un ou plusieurs fichiers. Chaque ligne du fichier est un document d'entrée distinct. Ce format est idéal pour les documents courts, tels que les SMS ou les publications sur les réseaux sociaux.	Un document par ligne

Un document par fichier

Avec le `one document per file` format, chaque fichier représente un document d'entrée.

Un document par ligne

Avec le `One document per line` format, chaque document est placé sur une ligne séparée et aucun en-tête n'est utilisé. L'étiquette n'est pas incluse sur chaque ligne (puisque vous ne connaissez pas encore l'étiquette du document). Chaque ligne du fichier (la fin du document individuel) doit se terminer par un flux de ligne (LF,\n), un retour en chariot (CR, \ r), ou les deux (CRLF, \ r\n). N'utilisez pas le séparateur de ligne UTF-8 (u+2028) pour terminer une ligne.

L'exemple suivant montre le format du fichier d'entrée.

```
Text of document 1 \n
Text of document 2 \n
Text of document 3 \n
Text of document 4 \n
```

Quel que soit le format, utilisez le codage UTF-8 pour les fichiers texte. Après avoir préparé les fichiers, placez-les dans le compartiment S3 que vous utilisez pour les données d'entrée.

Lorsque vous démarrez une tâche de classification, vous spécifiez cet emplacement Amazon S3 pour vos données d'entrée. L'URI doit se trouver dans la même région que le point de terminaison de l'API que vous appelez. L'URI peut pointer vers un seul fichier (comme lors de l'utilisation de la méthode « un document par ligne »), ou il peut être le préfixe d'un ensemble de fichiers de données.

Par exemple, si vous utilisez l'URI`S3://bucketName/prefix`, si le préfixe est un fichier unique, Amazon Comprehend utilise ce fichier en entrée. Si plusieurs fichiers commencent par le préfixe, Amazon Comprehend les utilise tous comme entrée.

Accordez à Amazon Comprehend l'accès au compartiment S3 qui contient votre collection de documents et vos fichiers de sortie. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Tâches d'analyse pour une classification personnalisée (console)

Après avoir créé et entraîné un classificateur de documents personnalisé, vous pouvez utiliser la console pour exécuter des tâches de classification personnalisées avec le modèle.

Pour créer une tâche de classification personnalisée (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Tâches d'analyse, puis sélectionnez Créer une tâche.

3. Donnez un nom à la tâche de classification. Le nom doit être propre à votre compte et à votre région actuelle.
4. Sous Type d'analyse, sélectionnez Classification personnalisée.
5. Dans Sélectionner un classificateur, choisissez le classificateur personnalisé à utiliser.
6. (Facultatif) Si vous choisissez de chiffrer les données utilisées par Amazon Comprehend lors du traitement de votre tâche, choisissez Job encryption. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

   - Si vous utilisez une clé associée au compte actuel, choisissez l'ID de clé pour l'ID de clé KMS.
   - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'ID de clé sous ARN de la clé KMS.

   > **Note**
   >
   > Pour plus d'informations sur la création et l'utilisation des clés KMS et le chiffrement associé, consultez la section Service de gestion des clés (KMS).

7. Sous Données d'entrée, entrez l'emplacement du compartiment Amazon S3 qui contient vos documents d'entrée ou naviguez jusqu'à celui-ci en choisissant Browse S3. Ce compartiment doit se trouver dans la même région que l'API que vous appelez. Le rôle IAM que vous utilisez pour les autorisations d'accès pour la tâche de classification doit disposer d'autorisations de lecture pour le compartiment S3.

   Pour obtenir le plus haut niveau de précision lors de l'entraînement d'un modèle, faites correspondre le type d'entrée au type de modèle de classificateur. La tâche de classification renvoie un avertissement si vous soumettez des documents natifs à un modèle de texte brut ou des documents en texte brut à un modèle de document natif. Pour de plus amples informations, veuillez consulter Modèles de classification des formations.
8. (Facultatif) Pour le format d'entrée, vous pouvez choisir le format des documents d'entrée. Le format peut être un document par fichier ou un document par ligne dans un seul fichier. Un document par ligne s'applique uniquement aux documents texte.
9. (Facultatif) En mode lecture de document, vous pouvez remplacer les actions d'extraction de texte par défaut. Pour de plus amples informations, veuillez consulter Configuration des options d'extraction de texte.
10. Sous Données de sortie, entrez l'emplacement du compartiment Amazon S3 dans lequel Amazon Comprehend doit écrire les données de sortie de la tâche ou accédez à celui-ci en

choisissant Browse S3. Ce compartiment doit se trouver dans la même région que l'API que vous appelez. Le rôle IAM que vous utilisez pour les autorisations d'accès pour la tâche de classification doit disposer d'autorisations d'écriture pour le compartiment S3.

11. (Facultatif) Si vous choisissez de chiffrer le résultat de sortie de votre tâche, choisissez Chiffrement. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

    - Si vous utilisez une clé associée au compte actuel, choisissez l'alias de clé ou l'ID pour l'ID de clé KMS.
    - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'alias ou de l'ID de clé sous ID de clé KMS.

12. (Facultatif) Pour lancer vos ressources dans Amazon Comprehend à partir d'un VPC, entrez l'ID du VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseau (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.
    2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

    > **Note**
    >
    > Lorsque vous utilisez un VPC avec votre tâche de classification, le VPC `DataAccessRole` utilisé pour les opérations de création et de démarrage doit accorder des autorisations au VPC qui accède au compartiment de sortie.

13. Choisissez Créer une tâche pour créer la tâche de classification de documents.

Tâches d'analyse pour une classification personnalisée (API)

Après avoir créé et entraîné un classificateur de documents personnalisé, vous pouvez l'utiliser pour exécuter des tâches d'analyse.

Utilisez cette StartDocumentClassificationJobopération pour commencer à classer les documents non étiquetés. Vous spécifiez le compartiment S3 qui contient les documents d'entrée, le compartiment S3 pour les documents de sortie et le classificateur à utiliser.

Pour obtenir le plus haut niveau de précision lors de l'entraînement d'un modèle, faites correspondre le type d'entrée au type de modèle de classificateur. La tâche de classification renvoie un avertissement si vous soumettez des documents natifs à un modèle de texte brut ou des documents en texte brut à un modèle de document natif. Pour de plus amples informations, veuillez consulter Modèles de classification des formations.

StartDocumentClassificationJobest asynchrone. Une fois que vous avez commencé la tâche, utilisez l'DescribeDocumentClassificationJobopération pour suivre sa progression. Lorsque le `Status` champ de la réponse s'afficheCOMPLETED, vous pouvez accéder à la sortie à l'emplacement que vous avez spécifié.

Rubriques

- À l'aide du AWS Command Line Interface
- Utilisation du SDK AWS SDK pour Java or pour Python

À l'aide du AWS Command Line Interface

Les exemples suivants illustrent l'`StartDocumentClassificationJob`opération et un autre classificateur personnalisé APIs avec le AWS CLI.

Les exemples suivants utilisent le format de commande pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Exécutez une tâche de classification personnalisée à l'aide de l'`StartDocumentClassificationJob`opération.

```
aws comprehend start-document-classification-job \
     --region region \
     --document-classifier-arn arn:aws:comprehend:region:account number:document-
classifier/testDelete \
     --input-data-config S3Uri=s3://S3Bucket/docclass/file
 name,InputFormat=ONE_DOC_PER_LINE \
     --output-data-config S3Uri=s3://S3Bucket/output \
     --data-access-role-arn arn:aws:iam::account number:role/resource name
```

Obtenez des informations sur un classificateur personnalisé avec l'identifiant de la tâche à l'aide de l'`DescribeDocumentClassificationJob`opération.

```
aws comprehend describe-document-classification-job \
     --region region \
     --job-id job id
```

Répertoriez toutes les tâches de classification personnalisées de votre compte à l'aide de `ListDocumentClassificationJobs` cette opération.

```
aws comprehend list-document-classification-jobs
     --region region
```

Utilisation du SDK AWS SDK pour Java or pour Python

Pour obtenir des exemples de SDK expliquant comment démarrer une tâche de classificateur personnalisée, consultez. Utilisation `StartDocumentClassificationJob` avec un AWS SDK ou une CLI

Sorties pour les tâches d'analyse asynchrones

Une fois qu'une tâche d'analyse est terminée, elle stocke les résultats dans le compartiment S3 que vous avez spécifié dans la demande.

Sorties pour entrées de texte

Quel que soit le format des documents de saisie de texte (multi-classes ou multi-étiquettes), le résultat de la tâche consiste en un seul fichier nommé. `output.tar.gz` Il s'agit d'un fichier d'archive compressé qui contient un fichier texte avec le résultat.

Sortie multiclasse

Lorsque vous utilisez un classificateur entraîné en mode multi-classes, vos résultats s'affichent. `classes` Chacune d'entre elles `classes` est la classe utilisée pour créer l'ensemble de catégories lors de l'entraînement de votre classificateur.

Pour plus de détails sur ces champs de sortie, consultez ClassifyDocumentle manuel Amazon Comprehend API Reference.

Les exemples suivants utilisent les classes mutuellement exclusives suivantes.

```
DOCUMENTARY
```

```
SCIENCE_FICTION
ROMANTIC_COMEDY
SERIOUS_DRAMA
OTHER
```

Si le format de vos données d'entrée est d'un document par ligne, le fichier de sortie contient une ligne pour chaque ligne de l'entrée. Chaque ligne inclut le nom du fichier, le numéro de ligne de base zéro de la ligne d'entrée et la ou les classes présentes dans le document. Cela se termine par la certitude qu'Amazon Comprehend a que l'instance individuelle a été correctement classée.

Par exemple :

```
{"File": "file1.txt", "Line": "0", "Classes": [{"Name": "Documentary", "Score":
 0.8642}, {"Name": "Other", "Score": 0.0381}, {"Name": "Serious_Drama", "Score":
 0.0372}]}
{"File": "file1.txt", "Line": "1", "Classes": [{"Name": "Science_Fiction", "Score":
 0.5}, {"Name": "Science_Fiction", "Score": 0.0381}, {"Name": "Science_Fiction",
 "Score": 0.0372}]}
{"File": "file2.txt", "Line": "2", "Classes": [{"Name": "Documentary", "Score": 0.1},
 {"Name": "Documentary", "Score": 0.0381}, {"Name": "Documentary", "Score": 0.0372}]}
{"File": "file2.txt", "Line": "3", "Classes": [{"Name": "Serious_Drama", "Score":
 0.3141}, {"Name": "Other", "Score": 0.0381}, {"Name": "Other", "Score": 0.0372}]}
```

Si le format de vos données d'entrée est un document par fichier, le fichier de sortie contient une ligne pour chaque document. Chaque ligne contient le nom du fichier et la ou les classes présentes dans le document. Cela se termine par la certitude qu'Amazon Comprehend a classé l'instance individuelle avec précision.

Par exemple :

```
{"File": "file0.txt", "Classes": [{"Name": "Documentary", "Score": 0.8642}, {"Name":
 "Other", "Score": 0.0381}, {"Name": "Serious_Drama", "Score": 0.0372}]}
{"File": "file1.txt", "Classes": [{"Name": "Science_Fiction", "Score": 0.5}, {"Name":
 "Science_Fiction", "Score": 0.0381}, {"Name": "Science_Fiction", "Score": 0.0372}]}
{"File": "file2.txt", "Classes": [{"Name": "Documentary", "Score": 0.1}, {"Name":
 "Documentary", "Score": 0.0381}, {"Name": "Domentary", "Score": 0.0372}]}
{"File": "file3.txt", "Classes": [{"Name": "Serious_Drama", "Score": 0.3141}, {"Name":
 "Other", "Score": 0.0381}, {"Name": "Other", "Score": 0.0372}]}
```

Sortie multi-étiquettes

Lorsque vous utilisez un classificateur entraîné en mode multi-étiquettes, vos résultats s'affichent. `labels` Chacune de ces étiquettes `labels` est utilisée pour créer l'ensemble de catégories lors de l'entraînement de votre classificateur.

Les exemples suivants utilisent ces étiquettes uniques.

```
SCIENCE_FICTION
ACTION
DRAMA
COMEDY
ROMANCE
```

Si le format de vos données d'entrée est d'un document par ligne, le fichier de sortie contient une ligne pour chaque ligne de l'entrée. Chaque ligne inclut le nom du fichier, le numéro de ligne de base zéro de la ligne d'entrée et la ou les classes présentes dans le document. Cela se termine par la certitude qu'Amazon Comprehend a que l'instance individuelle a été correctement classée.

Par exemple :

```
{"File": "file1.txt", "Line": "0", "Labels": [{"Name": "Action", "Score": 0.8642},
 {"Name": "Drama", "Score": 0.650}, {"Name": "Science Fiction", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "1", "Labels": [{"Name": "Comedy", "Score": 0.5},
 {"Name": "Action", "Score": 0.0381}, {"Name": "Drama", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "2", "Labels": [{"Name": "Action", "Score": 0.9934},
 {"Name": "Drama", "Score": 0.0381}, {"Name": "Action", "Score": 0.0372}]}
{"File": "file1.txt", "Line": "3", "Labels": [{"Name": "Romance", "Score": 0.9845},
 {"Name": "Comedy", "Score": 0.8756}, {"Name": "Drama", "Score": 0.7723}, {"Name":
 "Science_Fiction", "Score": 0.6157}]}
```

Si le format de vos données d'entrée est un document par fichier, le fichier de sortie contient une ligne pour chaque document. Chaque ligne contient le nom du fichier et la ou les classes présentes dans le document. Cela se termine par la certitude qu'Amazon Comprehend a classé l'instance individuelle avec précision.

Par exemple :

```
{"File": "file0.txt", "Labels": [{"Name": "Action", "Score": 0.8642}, {"Name": "Drama",
 "Score": 0.650}, {"Name": "Science Fiction", "Score": 0.0372}]}
{"File": "file1.txt", "Labels": [{"Name": "Comedy", "Score": 0.5}, {"Name": "Action",
 "Score": 0.0381}, {"Name": "Drama", "Score": 0.0372}]}
```

```
{"File": "file2.txt", "Labels": [{"Name": "Action", "Score": 0.9934}, {"Name": "Drama",
 "Score": 0.0381}, {"Name": "Action", "Score": 0.0372}]}
{"File": "file3.txt", "Labels": [{"Name": "Romance", "Score": 0.9845}, {"Name":
 "Comedy", "Score": 0.8756}, {"Name": "Drama", "Score": 0.7723}, {"Name":
 "Science_Fiction", "Score": 0.6157}]}
```

Sorties pour documents d'entrée semi-structurés

Pour les documents d'entrée semi-structurés, la sortie peut inclure les champs supplémentaires suivants :

- DocumentMetadata — Informations d'extraction relatives au document. Les métadonnées incluent une liste de pages du document, avec le nombre de caractères extraits de chaque page. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- DocumentType — Type de document pour chaque page du document d'entrée. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- Erreurs : erreurs au niveau de la page détectées par le système lors du traitement du document d'entrée. Le champ est vide si le système n'a détecté aucune erreur.

Pour plus de détails sur ces champs de sortie, consultez ClassifyDocumentle manuel Amazon Comprehend API Reference.

L'exemple suivant montre la sortie d'un fichier PDF numérisé de deux pages.

```
[{ #First page output
    "Classes": [
        {
            "Name": "__label__2 ",
            "Score": 0.9993996620178223
        },
        {
            "Name": "__label__3 ",
            "Score": 0.0004330444789957255
        }
    ],
    "DocumentMetadata": {
        "PageNumber": 1,
        "Pages": 2
    },
```

```
    "DocumentType": "ScannedPDF",
    "File": "file.pdf",
    "Version": "VERSION_NUMBER"
},
#Second page output
{
    "Classes": [
        {
            "Name": "__label__2 ",
            "Score": 0.9993996620178223
        },
        {
            "Name": "__label__3 ",
            "Score": 0.0004330444789957255
        }
    ],
    "DocumentMetadata": {
        "PageNumber": 2,
        "Pages": 2
    },
    "DocumentType": "ScannedPDF",
    "File": "file.pdf",
    "Version": "VERSION_NUMBER"
}]
```

Reconnaissance d'entités personnalisée

La reconnaissance personnalisée des entités étend les fonctionnalités d'Amazon Comprehend en vous aidant à identifier vos nouveaux types d'entités spécifiques qui ne figurent pas dans les types d'entités génériques prédéfinis. Cela signifie que vous pouvez analyser des documents et extraire des entités telles que des codes de produit ou des entités spécifiques à l'entreprise qui répondent à vos besoins particuliers.

Construire vous-même un outil de reconnaissance d'entités personnalisé précis peut s'avérer un processus complexe, qui nécessite la préparation de grands ensembles de documents de formation annotés manuellement et la sélection des algorithmes et paramètres appropriés pour l'entraînement des modèles. Amazon Comprehend contribue à réduire la complexité en fournissant des annotations automatiques et en développant des modèles afin de créer un modèle de reconnaissance d'entité personnalisé.

La création d'un modèle de reconnaissance d'entités personnalisé est une approche plus efficace que l'utilisation de correspondances de chaînes ou d'expressions régulières pour extraire des entités de documents. Par exemple, pour extraire les noms d'ENGINEER dans un document, il est difficile d'énumérer tous les noms possibles. De plus, sans contexte, il est difficile de faire la distinction entre les noms ENGINEER et ANALYST. Un modèle de reconnaissance d'entité personnalisé peut connaître le contexte dans lequel ces noms sont susceptibles d'apparaître. En outre, la correspondance de chaînes ne détectera pas les entités présentant des fautes de frappe ou respectant de nouvelles conventions de dénomination, alors que cela est possible à l'aide d'un modèle personnalisé.

Deux options s'offrent à vous pour créer un modèle personnalisé :

1. Annotations : fournissez un ensemble de données contenant des entités annotées pour l'entraînement des modèles.
2. Listes d'entités (texte brut uniquement) : fournissez une liste d'entités et leur étiquette de type (par exemple, un ensemble de documents non annotés contenant ces entités pour l'apprentissage des modèles). `PRODUCT_CODES`

Lorsque vous créez un outil de reconnaissance d'entités personnalisé à l'aide de fichiers PDF annotés, vous pouvez l'utiliser avec différents formats de fichiers d'entrée : texte brut, fichiers image (JPG, PNG, TIFF), fichiers PDF et documents Word, sans prétraitement ni aplatissement

du document. Amazon Comprehend ne prend pas en charge l'annotation de fichiers image ou de documents Word.

> **Note**
>
> Un outil de reconnaissance d'entités personnalisé utilisant des fichiers PDF annotés ne prend en charge que les documents en anglais.

Vous pouvez entraîner un modèle sur un maximum de 25 entités personnalisées à la fois. Pour plus de détails, consultez la page Directives et quotas.

Une fois votre modèle entraîné, vous pouvez l'utiliser pour la détection d'entités en temps réel et dans le cadre de tâches de détection d'entités.

Rubriques

- Préparation des données de formation du système de reconnaissance d'entités
- Former des modèles de reconnaissance d'entités personnalisés
- Exécution d'une analyse de reconnaissance personnalisée en temps réel
- Exécution de tâches d'analyse pour une reconnaissance d'entités personnalisée

Préparation des données de formation du système de reconnaissance d'entités

Pour former un modèle personnalisé de reconnaissance d'entités efficace, il est important de fournir au formateur des données de haute qualité en entrée. Sans données fiables, le modèle n'apprendra pas à identifier correctement les entités.

Vous pouvez choisir l'une des deux méthodes suivantes pour fournir des données à Amazon Comprehend afin de créer un modèle de reconnaissance d'entité personnalisé :

- Liste des entités : répertorie les entités spécifiques afin qu'Amazon Comprehend puisse s'entraîner à identifier vos entités personnalisées. Remarque : Les listes d'entités ne peuvent être utilisées que pour les documents en texte brut.
- Annotations — Fournit l'emplacement de vos entités dans un certain nombre de documents afin qu'Amazon Comprehend puisse s'entraîner à la fois sur l'entité et sur son contexte. Pour créer

un modèle permettant d'analyser des fichiers image ou des documents Word PDFs, vous devez entraîner votre outil de reconnaissance à l'aide d'annotations PDF.

Dans les deux cas, Amazon Comprehend découvre le type de documents et le contexte dans lequel les entités apparaissent et crée un outil de reconnaissance capable de généraliser pour détecter les nouvelles entités lorsque vous analysez des documents.

Lorsque vous créez un modèle personnalisé (ou que vous entraînez une nouvelle version), vous pouvez fournir un jeu de données de test. Si vous ne fournissez pas de données de test, Amazon Comprehend réserve 10 % des documents d'entrée pour tester le modèle. Amazon Comprehend entraîne le modèle avec les autres documents.

Si vous fournissez un jeu de données de test pour votre ensemble d'entraînement aux annotations, les données de test doivent inclure au moins une annotation pour chacun des types d'entités spécifiés dans la demande de création.

Rubriques

- Quand utiliser des annotations plutôt que des listes d'entités
- Listes d'entités (texte brut uniquement)
- Annotations

Quand utiliser des annotations plutôt que des listes d'entités

La création d'annotations demande plus de travail que la création d'une liste d'entités, mais le modèle obtenu peut être nettement plus précis. L'utilisation d'une liste d'entités est plus rapide et demande moins de travail, mais les résultats sont moins précis et moins précis. Cela est dû au fait que les annotations fournissent plus de contexte à Amazon Comprehend lors de l'entraînement du modèle. Sans ce contexte, Amazon Comprehend aura un plus grand nombre de faux positifs lorsqu'il tentera d'identifier les entités.

Il existe des scénarios dans lesquels il est plus judicieux sur le plan commercial d'éviter les dépenses et la charge de travail accrues liées à l'utilisation des annotations. Par exemple, le nom John Johnson est important pour votre recherche, mais il n'est pas pertinent de savoir s'il s'agit de la personne exacte. Ou bien, lorsque vous utilisez la liste d'entités, les indicateurs sont suffisamment bons pour vous fournir les résultats de reconnaissance dont vous avez besoin. Dans de tels cas, l'utilisation d'une liste d'entités peut être le choix le plus efficace.

Nous recommandons d'utiliser le mode annotations dans les cas suivants :

- Si vous prévoyez d'effectuer des inférences pour des fichiers image ou PDFs des documents Word. Dans ce scénario, vous entraînez un modèle à l'aide de fichiers PDF annotés et vous l'utilisez pour exécuter des tâches d'inférence pour des fichiers image et des documents Word. PDFs
- Quand la signification des entités peut être ambiguë et dépendre du contexte. Par exemple, le terme Amazon peut désigner le fleuve au Brésil ou le détaillant en ligne Amazon.com. Lorsque vous créez un outil de reconnaissance d'entités personnalisé pour identifier des entités commerciales telles qu'Amazon, vous devez utiliser des annotations plutôt qu'une liste d'entités, car cette méthode permet de mieux utiliser le contexte pour rechercher des entités.
- Lorsque vous êtes à l'aise pour configurer un processus d'acquisition d'annotations, ce qui peut nécessiter un certain effort.

Nous vous recommandons d'utiliser une liste d'entités dans les cas suivants :

- Lorsque vous avez déjà une liste d'entités ou lorsqu'il est relativement facile de composer une liste complète d'entités. Si vous utilisez une liste d'entités, celle-ci doit être complète ou au moins couvrir la majorité des entités valides susceptibles de figurer dans les documents que vous fournissez pour la formation.
- Pour les nouveaux utilisateurs, il est généralement recommandé d'utiliser une liste d'entités, car cela demande moins d'efforts que la création d'annotations. Cependant, il est important de noter que le modèle entraîné n'est peut-être pas aussi précis que si vous utilisiez des annotations.

Listes d'entités (texte brut uniquement)

Pour entraîner un modèle à l'aide d'une liste d'entités, vous devez fournir deux informations : une liste des noms d'entités avec les types d'entités personnalisés correspondants et une collection de documents non annotés dans lesquels vous vous attendez à ce que vos entités apparaissent.

Lorsque vous fournissez une liste d'entités, Amazon Comprehend utilise un algorithme intelligent pour détecter les occurrences de l'entité dans les documents afin de servir de base à la formation du modèle de reconnaissance d'entités personnalisé.

Pour les listes d'entités, fournissez au moins 25 entités correspondantes par type d'entité dans la liste d'entités.

Une liste d'entités pour la reconnaissance d'entités personnalisée nécessite un fichier de valeurs séparées par des virgules (CSV), avec les colonnes suivantes :

- Texte — Le texte d'un exemple d'entrée tel qu'il apparaît dans le corpus du document d'accompagnement.
- Type : type d'entité défini par le client. Les types d'entités doivent être constitués d'une chaîne séparée par des majuscules et soulignés, telle que MANAGER ou SENIOR_MANAGER. Jusqu'à 25 types d'entités peuvent être formés par modèle.

Le fichier `documents.txt` contient quatre lignes :

```
Jo Brown is an engineer in the high tech industry.
John Doe has been a engineer for 14 years.
Emilio Johnson is a judge on the Washington Supreme Court.
Our latest new employee, Jane Smith, has been a manager in the industry for 4 years.
```

Le fichier CSV contenant la liste des entités comporte les lignes suivantes :

```
Text, Type
Jo Brown, ENGINEER
John Doe, ENGINEER
Jane Smith, MANAGER
```

> **Note**
>
> Dans la liste des entités, l'entrée pour Emilio Johnson n'est pas présente car elle ne contient ni l'entité ENGINEER ni l'entité MANAGER.

Création de vos fichiers de données

Il est important que votre liste d'entités figure dans un fichier CSV correctement configuré afin de minimiser les risques de problèmes avec votre fichier de liste d'entités. Pour configurer manuellement votre fichier CSV, les conditions suivantes doivent être remplies :

- Le codage UTF-8 doit être explicitement spécifié, même s'il est utilisé par défaut dans la plupart des cas.
- Il doit inclure les noms des colonnes : `Type` et`Text`.

Nous recommandons vivement de générer les fichiers d'entrée CSV par programmation afin d'éviter d'éventuels problèmes.

L'exemple suivant utilise Python pour générer un fichier CSV pour les annotations présentées ci-dessus :

```
import csv
with open("./entitylist/entitylist.csv", "w", encoding="utf-8") as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(["Text", "Type"])
    csv_writer.writerow(["Jo Brown", " ENGINEER"])
    csv_writer.writerow(["John Doe", " ENGINEER"])
    csv_writer.writerow(["Jane Smith", " MANAGER"])
```

Bonnes pratiques

Plusieurs éléments doivent être pris en compte pour obtenir le meilleur résultat lorsque vous utilisez une liste d'entités, notamment :

- L'ordre des entités dans votre liste n'a aucun effet sur l'entraînement des modèles.
- Utilisez des éléments de liste d'entités qui couvrent 80 % à 100 % des exemples d'entités positives mentionnés dans le corpus de documents non annoté.
- Évitez les exemples d'entités qui correspondent à des entités non présentes dans le corpus de documents en supprimant les mots et expressions courants. Même une poignée de correspondances incorrectes peuvent affecter de manière significative la précision du modèle obtenu. Par exemple, un mot tel que « dans la liste des entités » se traduira par un grand nombre de correspondances qui ne correspondent probablement pas aux entités que vous recherchez et affectera donc de manière significative votre précision.
- Les données d'entrée ne doivent pas contenir de doublons. La présence d'échantillons dupliqués peut entraîner la contamination du set de test et donc affecter négativement le processus de formation, les métriques du modèle et le comportement.
- Fournissez des documents qui ressemblent le plus possible à des cas d'utilisation réels. N'utilisez pas de données sur les jouets ou de données synthétisées pour les systèmes de production. Les données d'entrée doivent être aussi diverses que possible pour éviter un ajustement excessif et aider le modèle sous-jacent à mieux généraliser sur des exemples réels.
- La liste des entités distingue les majuscules et minuscules, et les expressions régulières ne sont actuellement pas prises en charge. Cependant, le modèle entraîné peut souvent encore

reconnaître les entités même si elles ne correspondent pas exactement au cas indiqué dans la liste des entités.

- Si une entité est une sous-chaîne d'une autre entité (telle que « Smith » et « Jane Smith »), indiquez les deux dans la liste des entités.

Des suggestions supplémentaires peuvent être trouvées sur Améliorer les performances du système de reconnaissance d'entités personnalisé

Annotations

Les annotations étiquettent les entités dans leur contexte en associant vos types d'entités personnalisés aux emplacements où ils apparaissent dans vos documents de formation.

En soumettant des annotations avec vos documents, vous pouvez augmenter la précision du modèle. Avec les annotations, vous ne vous contentez pas de fournir l'emplacement de l'entité que vous recherchez, mais vous fournissez également un contexte plus précis à l'entité personnalisée que vous recherchez.

Par exemple, si vous recherchez le nom John Johnson, avec le type d'entité JUDGE, le fait de fournir votre annotation peut aider le modèle à apprendre que la personne que vous recherchez est un juge. S'il est capable d'utiliser le contexte, Amazon Comprehend ne trouvera pas de personnes nommées John Johnson qui sont des avocats ou des témoins. Sans fournir d'annotations, Amazon Comprehend créera sa propre version d'une annotation, mais ne sera pas aussi efficace pour inclure uniquement les juges. Fournir vos propres annotations peut vous aider à obtenir de meilleurs résultats et à générer des modèles capables de mieux tirer parti du contexte lors de l'extraction d'entités personnalisées.

Rubriques

- Nombre minimum d'annotations
- Bonnes pratiques en matière d'annotation
- Fichiers d'annotations en texte brut
- Fichiers d'annotation PDF
- Annotation de fichiers PDF

Nombre minimum d'annotations

Le nombre minimum de documents d'entrée et d'annotations requis pour entraîner un modèle dépend du type d'annotations.

Annotations au format PDF

Pour créer un modèle permettant d'analyser des fichiers image ou des documents Word PDFs, entraînez votre outil de reconnaissance à l'aide d'annotations PDF. Pour les annotations PDF, fournissez au moins 250 documents d'entrée et au moins 100 annotations par entité.

Si vous fournissez un jeu de données de test, les données de test doivent inclure au moins une annotation pour chacun des types d'entités spécifiés dans la demande de création.

Annotations en texte brut

Pour créer un modèle d'analyse de documents texte, vous pouvez entraîner votre outil de reconnaissance à l'aide d'annotations en texte brut.

Pour les annotations en texte brut, fournissez au moins trois documents d'entrée annotés et au moins 25 annotations par entité. Si vous fournissez moins de 50 annotations au total, Amazon Comprehend réserve plus de 10 % des documents d'entrée pour tester le modèle (sauf si vous avez fourni un ensemble de données de test dans la demande de formation). N'oubliez pas que la taille minimale du corpus de documents est de 5 Ko.

Si votre entrée ne contient que quelques documents de formation, il se peut que vous rencontriez une erreur indiquant que les données d'entrée de formation contiennent trop peu de documents mentionnant l'une des entités. Soumettez à nouveau le travail avec des documents supplémentaires mentionnant l'entité.

Si vous fournissez un jeu de données de test, les données de test doivent inclure au moins une annotation pour chacun des types d'entités spécifiés dans la demande de création.

Pour voir comment comparer un modèle à un petit ensemble de données, consultez Amazon Comprehend annonce des limites d'annotations plus basses pour la reconnaissance d'entités personnalisée sur le AWS site de blog.

Bonnes pratiques en matière d'annotation

Plusieurs éléments doivent être pris en compte pour obtenir les meilleurs résultats lors de l'utilisation d'annotations, notamment :

- Annotez vos données avec soin et vérifiez que vous annotez chaque mention de l'entité. Des annotations imprécises peuvent entraîner des résultats médiocres.
- Les données d'entrée ne doivent pas contenir de doublons, comme le double d'un PDF que vous allez annoter. La présence d'un échantillon dupliqué peut entraîner la contamination du set de test et affecter négativement le processus de formation, les paramètres du modèle et le comportement du modèle.
- Assurez-vous que tous vos documents sont annotés et que les documents sans annotations sont dus à l'absence d'entités légitimes et non à une négligence. Par exemple, si vous avez un document qui dit « J Doe est ingénieur depuis 14 ans », vous devez également fournir une annotation pour « J Doe » et « John Doe ». Si vous ne le faites pas, le modèle risque de ne pas reconnaître « J Doe » comme ENGINEER. Cela doit être cohérent au sein d'un même document et d'un document à l'autre.
- En général, un plus grand nombre d'annotations permet d'obtenir de meilleurs résultats.
- Vous pouvez entraîner un modèle avec un minimum de documents et d'annotations, mais l'ajout de données améliore généralement le modèle. Nous recommandons d'augmenter le volume de données annotées de 10 % pour augmenter la précision du modèle. Vous pouvez exécuter l'inférence sur un jeu de données de test qui reste inchangé et peut être testé par différentes versions de modèle. Vous pouvez ensuite comparer les mesures pour les versions successives du modèle.
- Fournissez des documents qui ressemblent le plus possible à des cas d'utilisation réels. Les données synthétisées avec des modèles répétitifs doivent être évitées. Les données d'entrée doivent être aussi diverses que possible afin d'éviter un ajustement excessif et d'aider le modèle sous-jacent à mieux généraliser sur des exemples réels.
- Il est important que les documents soient diversifiés en termes de nombre de mots. Par exemple, si tous les documents contenus dans les données d'entraînement sont courts, le modèle obtenu peut avoir des difficultés à prévoir les entités dans des documents plus longs.
- Essayez de distribuer les données pour l'entraînement de la même manière que celle que vous comptez utiliser lorsque vous détectez réellement vos entités personnalisées (temps d'inférence). Par exemple, au moment de l'inférence, si vous prévoyez de nous envoyer des documents ne contenant aucune entité, cela devrait également faire partie de votre ensemble de documents de formation.

Pour des suggestions supplémentaires, voir Améliorer les performances du système de reconnaissance d'entités personnalisé.

Fichiers d'annotations en texte brut

Pour les annotations en texte brut, vous créez un fichier de valeurs séparées par des virgules (CSV) qui contient une liste d'annotations. Le fichier CSV doit contenir les colonnes suivantes si le format d'entrée de votre fichier de formation est d'un document par ligne.

Fichier	Line	Commencer le décalage	Décalage final	Type
Nom du fichier contenant le document. Par exemple, si l'un des fichiers du document se trouve dans`s3://my-S3-bucket/test-files/documents.txt` , la valeur de la `File` colonne sera`documents.txt` . Vous devez inclure l'extension du fichier (dans ce cas `.txt` « ») dans le nom du fichier.	Numéro de ligne contenant l'entité. Omettez cette colonne si votre format d'entrée est un document par fichier.	Décalage de caractères dans le texte saisi (par rapport au début de la ligne) qui indique le point de départ de l'entité. Le premier caractère est à la position 0.	Le décalage de caractères dans le texte d'entrée qui indique où se termine l'entité.	Type d'entité défini par le client. Les types d'entités doivent être des chaînes majuscules séparées par des traits de soulignement. Nous vous recommandons d'utiliser des types d'entités descriptifs tels que `MANAGERSENIOR_MANAGER` , ou`PRODUCT_CODE` . Jusqu'à 25 types d'entités peuvent être formés par modèle.

Si le format d'entrée de votre fichier de formation est d'un document par fichier, vous omettez la colonne des numéros de ligne et les valeurs de décalage de début et de fin sont les décalages de l'entité depuis le début du document.

L'exemple suivant concerne un document par ligne. Le fichier `documents.txt` contient quatre lignes (lignes 0, 1, 2 et 3) :

```
Diego Ramirez is an engineer in the high tech industry.
Emilio Johnson has been an engineer for 14 years.
J Doe is a judge on the Washington Supreme Court.
Our latest new employee, Mateo Jackson, has been a manager in the industry for 4 years.
```

Le fichier CSV contenant la liste des annotations est le suivant :

```
File, Line, Begin Offset, End Offset, Type
documents.txt, 0, 0, 13, ENGINEER
documents.txt, 1, 0, 14, ENGINEER
documents.txt, 3, 25, 38, MANAGER
```

> **Note**
>
> Dans le fichier d'annotations, le numéro de ligne contenant l'entité commence par la ligne 0. Dans cet exemple, le fichier CSV ne contient aucune entrée pour la ligne 2 car il n'y a aucune entité dans la ligne 2 de`documents.txt`.

Création de vos fichiers de données

Il est important de placer vos annotations dans un fichier CSV correctement configuré afin de réduire le risque d'erreurs. Pour configurer manuellement votre fichier CSV, les conditions suivantes doivent être remplies :

- Le codage UTF-8 doit être explicitement spécifié, même s'il est utilisé par défaut dans la plupart des cas.
- La première ligne contient les en-têtes de colonne :`File`, `Line` (facultatif),, `Begin Offset`,`End Offset`,`Type`.

Nous vous recommandons vivement de générer les fichiers d'entrée CSV par programmation afin d'éviter d'éventuels problèmes.

L'exemple suivant utilise Python pour générer un fichier CSV pour les annotations présentées précédemment :

```
import csv
with open("./annotations/annotations.csv", "w", encoding="utf-8") as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(["File", "Line", "Begin Offset", "End Offset", "Type"])
    csv_writer.writerow(["documents.txt", 0, 0, 11, "ENGINEER"])
    csv_writer.writerow(["documents.txt", 1, 0, 5, "ENGINEER"])
    csv_writer.writerow(["documents.txt", 3, 25, 30, "MANAGER"])
```

Fichiers d'annotation PDF

Pour les annotations PDF, vous utilisez SageMaker AI Ground Truth pour créer un ensemble de données étiqueté dans un fichier manifeste augmenté. Ground Truth est un service d'étiquetage des données qui vous aide (ou aide le personnel que vous employez) à créer des ensembles de données de formation pour les modèles d'apprentissage automatique. Amazon Comprehend accepte les fichiers de manifeste augmentés comme données d'entraînement pour les modèles personnalisés. Vous pouvez fournir ces fichiers lorsque vous créez un outil de reconnaissance d'entités personnalisé à l'aide de la console Amazon Comprehend ou de CreateEntityRecognizerl'action API.

Vous pouvez utiliser le type de tâche intégré à Ground Truth, nommé Entity Recognition, pour créer une tâche d'étiquetage afin que les travailleurs identifient les entités dans le texte. Pour en savoir plus, consultez la section Named Entity Recognition dans le manuel Amazon SageMaker AI Developer Guide. Pour en savoir plus sur Amazon SageMaker Ground Truth, consultez Utiliser Amazon SageMaker AI Ground Truth pour étiqueter les données.

> **Note**
>
> À l'aide de Ground Truth, vous pouvez définir des étiquettes qui se chevauchent (texte que vous associez à plusieurs étiquettes). Toutefois, la reconnaissance des entités Amazon Comprehend ne prend pas en charge le chevauchement des étiquettes.

Les fichiers manifestes augmentés sont au format de lignes JSON. Dans ces fichiers, chaque ligne est un objet JSON complet qui contient un document de formation et ses étiquettes associées. L'exemple suivant est un fichier manifeste augmenté qui entraîne un outil de reconnaissance d'entités à détecter les professions des personnes mentionnées dans le texte :

```
{"source":"Diego Ramirez is an engineer in the high tech
 industry.","NamedEntityRecognitionDemo":{"annotations":{"entities":
[{"endOffset":13,"startOffset":0,"label":"ENGINEER"}],"labels":
```

```
[{"label":"ENGINEER"}]}},"NamedEntityRecognitionDemo-metadata":
{"entities":[{"confidence":0.92}],"job-name":"labeling-job/
namedentityrecognitiondemo","type":"groundtruth/text-span","creation-
date":"2020-05-14T21:45:27.175903","human-annotated":"yes"}}
{"source":"J Doe is a judge on the Washington Supreme
 Court.","NamedEntityRecognitionDemo":{"annotations":{"entities":
[{"endOffset":5,"startOffset":0,"label":"JUDGE"}],"labels":
[{"label":"JUDGE"}]}},"NamedEntityRecognitionDemo-metadata":
{"entities":[{"confidence":0.72}],"job-name":"labeling-job/
namedentityrecognitiondemo","type":"groundtruth/text-span","creation-
date":"2020-05-14T21:45:27.174910","human-annotated":"yes"}}
{"source":"Our latest new employee, Mateo Jackson, has been a manager in
 the industry for 4 years.","NamedEntityRecognitionDemo":{"annotations":
{"entities":[{"endOffset":38,"startOffset":26,"label":"MANAGER"}],"labels":
[{"label":"MANAGER"}]}},"NamedEntityRecognitionDemo-metadata":
{"entities":[{"confidence":0.91}],"job-name":"labeling-job/
namedentityrecognitiondemo","type":"groundtruth/text-span","creation-
date":"2020-05-14T21:45:27.174035","human-annotated":"yes"}}
```


Chaque ligne de ce fichier de lignes JSON est un objet JSON complet, dont les attributs incluent le texte du document, les annotations et les autres métadonnées de Ground Truth. L'exemple suivant est un seul objet JSON dans le fichier manifeste augmenté, mais il est formaté pour en faciliter la lisibilité :


```
{
  "source": "Diego Ramirez is an engineer in the high tech industry.",
  "NamedEntityRecognitionDemo": {
    "annotations": {
      "entities": [
        {
          "endOffset": 13,
          "startOffset": 0,
          "label": "ENGINEER"
        }
      ],
      "labels": [
        {
          "label": "ENGINEER"
        }
      ]
    }
  },
  "NamedEntityRecognitionDemo-metadata": {
```

```
    "entities": [
      {
        "confidence": 0.92
      }
    ],
    "job-name": "labeling-job/namedentityrecognitiondemo",
    "type": "groundtruth/text-span",
    "creation-date": "2020-05-14T21:45:27.175903",
    "human-annotated": "yes"
  }
}
```

Dans cet exemple, l'`source`attribut fournit le texte du document de formation et l'`NamedEntityRecognitionDemo`attribut fournit les annotations pour les entités du texte. Le nom de l'`NamedEntityRecognitionDemo`attribut est arbitraire, et vous indiquez le nom de votre choix lorsque vous définissez la tâche d'étiquetage dans Ground Truth.

Dans cet exemple, l'`NamedEntityRecognitionDemo`attribut est le nom de l'attribut label, qui fournit les labels qu'un collaborateur de Ground Truth attribue aux données de formation. Lorsque vous fournissez vos données de formation à Amazon Comprehend, vous devez spécifier un ou plusieurs noms d'attributs d'étiquette. Le nombre de noms d'attributs que vous spécifiez varie selon que votre fichier manifeste augmenté est le résultat d'une tâche d'étiquetage unique ou d'une tâche d'étiquetage en chaîne.

Si votre fichier est le résultat d'une seule tâche d'étiquetage, spécifiez le nom d'attribut d'étiquette unique utilisé lors de la création de la tâche dans Ground Truth.

Si votre fichier est le résultat d'une tâche d'étiquetage en chaîne, spécifiez le nom de l'attribut d'étiquette pour une ou plusieurs tâches de la chaîne. Chaque nom d'attribut d'étiquette fournit les annotations d'une tâche individuelle. Vous pouvez spécifier jusqu'à 5 de ces attributs pour les fichiers manifestes augmentés produits par des tâches d'étiquetage en chaîne.

Dans un fichier manifeste augmenté, le nom de l'attribut d'étiquette suit généralement la `source` clé. Si le fichier est le résultat d'une tâche chaînée, les noms d'attributs d'étiquette seront multiples. Lorsque vous fournissez vos données d'entraînement à Amazon Comprehend, fournissez uniquement les attributs contenant des annotations pertinentes pour votre modèle. Ne spécifiez pas les attributs qui se terminent par « -metadata ».

Pour plus d'informations sur les tâches d'étiquetage en chaîne et pour obtenir des exemples des résultats qu'elles produisent, consultez la section Chaining Labeling Jobs dans le manuel Amazon SageMaker AI Developer Guide.

Annotation de fichiers PDF

Avant de pouvoir annoter votre formation PDFs dans SageMaker AI Ground Truth, vous devez remplir les prérequis suivants :

- Installer python3.8.x
- Installer jq
- Installation de la AWS CLI

  Si vous utilisez la région us-east-1, vous pouvez ignorer l'installation de la AWS CLI car elle est déjà installée dans votre environnement Python. Dans ce cas, vous créez un environnement virtuel pour utiliser Python 3.8 dans AWS Cloud9.
- Configurez vos AWS informations d'identification
- Créez une équipe privée d'SageMaker IA Ground Truth pour prendre en charge les annotations

  Assurez-vous d'enregistrer le nom de l'équipe de travail que vous avez choisi dans votre nouveau personnel privé, tel que vous l'utiliserez lors de l'installation.

Rubriques

- Configuration de votre environnement
- Chargement d'un PDF dans un compartiment S3
- Création d'une tâche d'annotation
- Annoter avec SageMaker AI Ground Truth

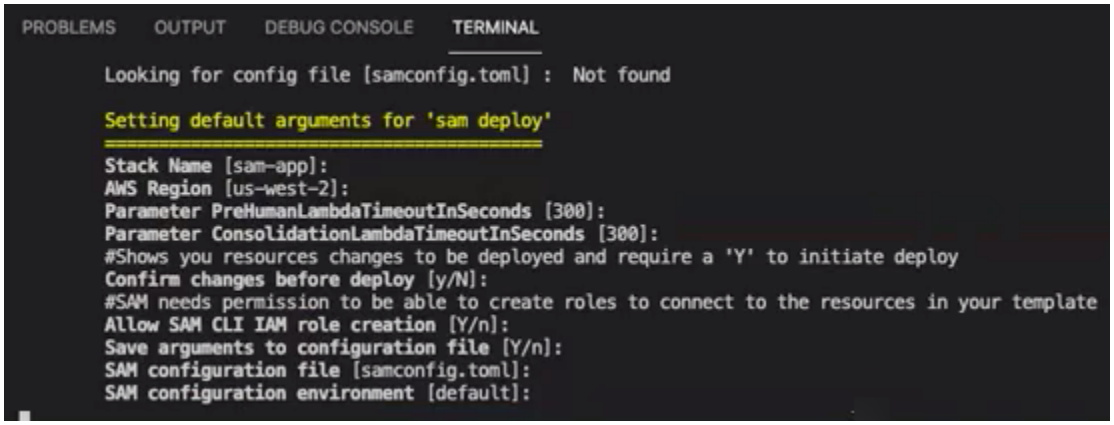
Configuration de votre environnement

1. Si vous utilisez Windows, installez Cygwin ; si vous utilisez Linux ou Mac, ignorez cette étape.
2. Téléchargez les artefacts d'annotation depuis GitHub. Décompressez le fichier.
3. Dans la fenêtre de votre terminal, accédez au dossier décompressé (amazon-comprehend-semi-structured- documents-annotation-tools-main).
4. Ce dossier inclut un choix de ceux `Makefiles` que vous pouvez exécuter pour installer les dépendances, configurer un environnement virtuel Python et déployer les ressources requises. Consultez le fichier readme pour faire votre choix.

5. L'option recommandée utilise une seule commande pour installer toutes les dépendances dans un environnement virtuel, crée la AWS CloudFormation pile à partir du modèle et déploie la pile sur vous Compte AWS avec des instructions interactives. Exécutez la commande suivante :

   ```
   make ready-and-deploy-guided
   ```

   Cette commande présente un ensemble d'options de configuration. Assurez-vous que vous Région AWS êtes correct. Pour tous les autres champs, vous pouvez accepter les valeurs par défaut ou renseigner des valeurs personnalisées. Si vous modifiez le nom de la AWS CloudFormation pile, notez-le selon vos besoins dans les étapes suivantes.

   La CloudFormation pile crée et gère les AWS lambdas, les rôles AWS IAM et les compartiments AWS S3 requis pour l'outil d'annotation.

   Vous pouvez consulter chacune de ces ressources sur la page de détails de la pile de la CloudFormation console.

6. La commande vous invite à démarrer le déploiement. CloudFormation crée toutes les ressources de la région spécifiée.

   Lorsque le statut de la CloudFormation pile passe à create-complete, les ressources sont prêtes à être utilisées.

Chargement d'un PDF dans un compartiment S3

Dans la section Configuration, vous avez déployé une CloudFormation pile qui crée un compartiment S3 nommé comprehend-semi-structured-documents-$ {AWS::Region}-${AWS::AccountId}. Vous chargez maintenant vos documents PDF source dans ce compartiment.

> **Note**
>
> Ce compartiment contient les données requises pour votre tâche d'étiquetage. La politique du rôle d'exécution Lambda autorise la fonction Lambda à accéder à ce compartiment.
> Vous pouvez trouver le nom du compartiment S3 dans les détails de la CloudFormation pile à l'aide de la clé « SemiStructuredDocumentsS3Bucket ».

1. Créez un nouveau dossier dans le compartiment S3. Nommez ce nouveau dossier « src ».
2. Ajoutez vos fichiers source PDF dans votre dossier « src ». Dans une étape ultérieure, vous annoterez ces fichiers pour entraîner votre outil de reconnaissance.
3. (Facultatif) Voici un exemple de AWS CLI que vous pouvez utiliser pour télécharger vos documents sources depuis un répertoire local vers un compartiment S3 :

   ```
   aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided/src/
   ```

   Ou, avec votre région et votre numéro de compte :

   ```
   aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided-Region-AccountID/src/
   ```
4. Vous disposez désormais d'une équipe privée d' SageMaker AI Ground Truth et vous avez téléchargé vos fichiers sources dans le compartiment S3, deploy-guided/src/ ; vous êtes prêt à commencer à annoter.

Création d'une tâche d'annotation

Le script comprehend-ssie-annotation-tool-cli.py contenu dans le `bin` répertoire est une simple commande wrapper qui rationalise la création d'une tâche d'étiquetage SageMaker AI Ground Truth. Le script python lit les documents sources de votre compartiment S3 et crée un fichier manifeste d'une seule page correspondant avec un document source par ligne. Le script crée ensuite une tâche d'étiquetage, qui nécessite le fichier manifeste comme entrée.

Le script python utilise le compartiment et la CloudFormation pile S3 que vous avez configurés dans la section Configuration. Les paramètres d'entrée requis pour le script sont les suivants :

- input-s3-path : URI S3 vers les documents source que vous avez chargés dans votre compartiment S3. olpPar exemple : `s3://deploy-guided/src/`. Vous pouvez également ajouter votre région et votre numéro de compte à ce chemin. olpPar exemple : `s3://deploy-guided-Region-AccountID/src/`.
- cfn-name : nom de la CloudFormation pile. Si vous avez utilisé la valeur par défaut pour le nom de la pile, votre nom de fichier cfn est sam-app.
- work-team-name: Le nom du personnel que vous avez créé lorsque vous avez créé le personnel privé dans SageMaker AI Ground Truth.
- job-name-prefix: le préfixe de la tâche d'étiquetage SageMaker AI Ground Truth. Notez que ce champ est limité à 29 caractères. Un horodatage est ajouté à cette valeur. olpPar exemple : `my-job-name-20210902T232116`.
- entity-types : les entités que vous souhaitez utiliser lors de votre travail d'étiquetage, séparées par des virgules. Cette liste doit inclure toutes les entités que vous souhaitez annoter dans votre jeu de données d'entraînement. La tâche d'étiquetage Ground Truth affiche uniquement ces entités afin que les annotateurs puissent étiqueter le contenu des documents PDF.

Pour afficher les arguments supplémentaires pris en charge par le script, utilisez l'`-h`option permettant d'afficher le contenu de l'aide.

- Exécutez le script suivant avec les paramètres d'entrée tels que décrits dans la liste précédente.

```
python bin/comprehend-ssie-annotation-tool-cli.py \
--input-s3-path s3://deploy-guided-Region-AccountID/src/ \
--cfn-name sam-app \
--work-team-name my-work-team-name \
--region us-east-1 \
--job-name-prefix my-job-name-20210902T232116 \
--entity-types "EntityA, EntityB, EntityC" \
--annotator-metadata "key=info,value=sample,key=Due Date,value=12/12/2021"
```

  Le script produit le résultat suivant :

```
Downloaded files to temp local directory /tmp/a1dc0c47-0f8c-42eb-9033-74a988ccc5aa
Deleted downloaded temp files from /tmp/a1dc0c47-0f8c-42eb-9033-74a988ccc5aa
```

```
Uploaded input manifest file to s3://comprehend-semi-structured-documents-
us-west-2-123456789012/input-manifest/my-job-name-20220203-labeling-
job-20220203T183118.manifest
Uploaded schema file to s3://comprehend-semi-structured-documents-us-
west-2-123456789012/comprehend-semi-structured-docs-ui-template/my-job-
name-20220203-labeling-job-20220203T183118/ui-template/schema.json
Uploaded template UI to s3://comprehend-semi-structured-documents-us-
west-2-123456789012/comprehend-semi-structured-docs-ui-template/my-job-
name-20220203-labeling-job-20220203T183118/ui-template/template-2021-04-15.liquid
Sagemaker GroundTruth Labeling Job submitted: arn:aws:sagemaker:us-
west-2:123456789012:labeling-job/my-job-name-20220203-labeling-job-20220203t183118
(amazon-comprehend-semi-structured-documents-annotation-tools-main)
 user@3c063014d632 amazon-comprehend-semi-structured-documents-annotation-tools-
main %
```

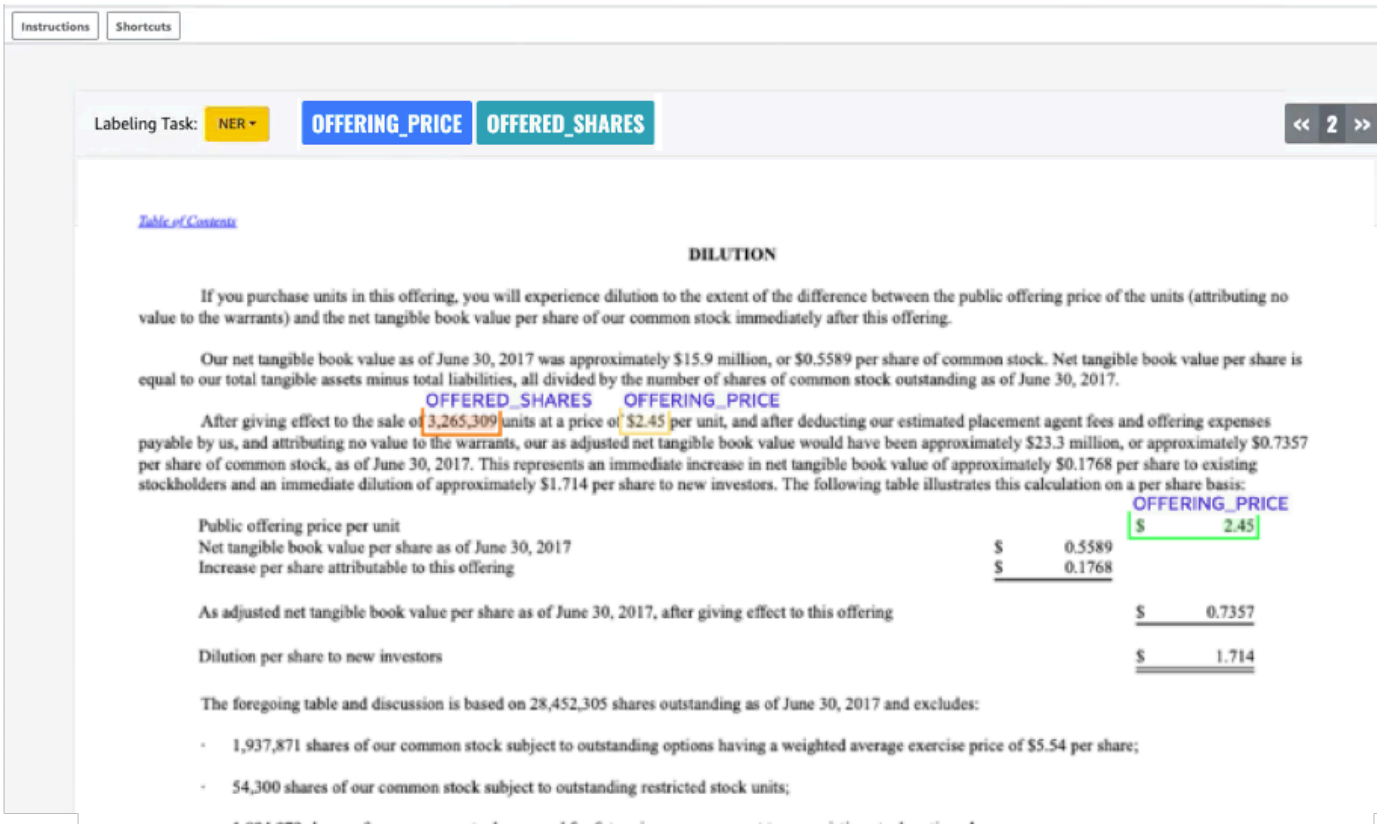
Annoter avec SageMaker AI Ground Truth

Maintenant que vous avez configuré les ressources requises et créé une tâche d'étiquetage, vous pouvez vous connecter au portail d'étiquetage et annoter votre PDFs.

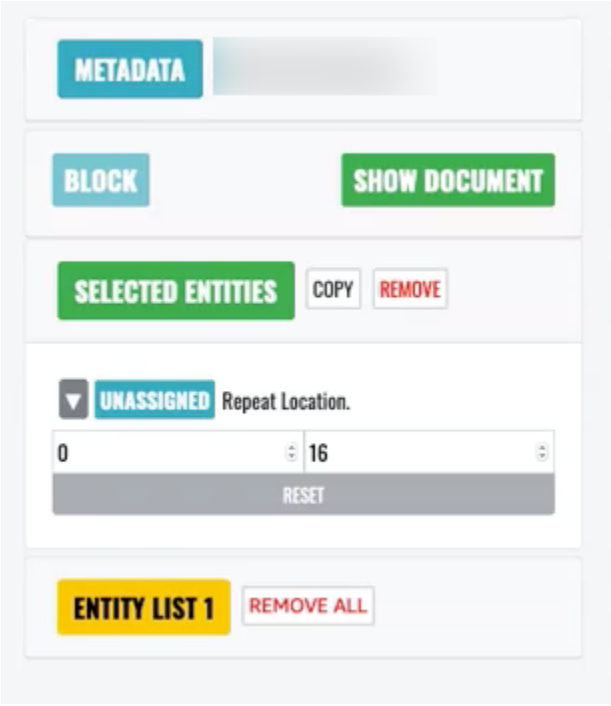
1. Connectez-vous à la console SageMaker AI à l'aide des navigateurs Web Chrome ou Firefox.
2. Sélectionnez Labeling workforce, puis Private.
3. Sous Résumé du personnel privé, sélectionnez l'URL de connexion au portail d'étiquetage que vous avez créée avec votre personnel privé. Connectez-vous avec les informations d'identification appropriées.

   Si aucune tâche n'est répertoriée, ne vous inquiétez pas : la mise à jour peut prendre un certain temps, en fonction du nombre de fichiers que vous avez chargés pour annotation.
4. Sélectionnez votre tâche et, dans le coin supérieur droit, choisissez Commencer à travailler pour ouvrir l'écran d'annotation.

   Vous verrez l'un de vos documents s'ouvrir dans l'écran d'annotation et, au-dessus, les types d'entités que vous avez fournis lors de la configuration. À droite de vos types d'entités, vous trouverez une flèche que vous pouvez utiliser pour parcourir vos documents.



Annotez le document ouvert. Vous pouvez également supprimer, annuler ou étiqueter automatiquement vos annotations sur chaque document ; ces options sont disponibles dans le panneau droit de l'outil d'annotation.



Pour utiliser la balise automatique, annotez une instance de l'une de vos entités ; toutes les autres instances de ce mot spécifique sont ensuite automatiquement annotées avec ce type d'entité.

Lorsque vous avez terminé, sélectionnez Soumettre en bas à droite, puis utilisez les flèches de navigation pour passer au document suivant. Répétez cette opération jusqu'à ce que vous ayez annoté tous vos PDFs.

Après avoir annoté tous les documents de formation, vous pouvez trouver les annotations au format JSON dans le compartiment Amazon S3 à cet emplacement :

```
/output/your labeling job name/annotations/
```

Le dossier de sortie contient également un fichier manifeste de sortie, qui répertorie toutes les annotations contenues dans vos documents de formation. Vous trouverez votre fichier manifeste de sortie à l'emplacement suivant.

```
/output/your labeling job name/manifests/
```

Former des modèles de reconnaissance d'entités personnalisés

Un outil de reconnaissance d'entités personnalisé identifie uniquement les types d'entités que vous incluez lorsque vous entraînez le modèle. Il n'inclut pas automatiquement les types d'entités prédéfinis. Si vous souhaitez également identifier les types d'entités prédéfinis, tels que LOCATION, DATE ou PERSON, vous devez fournir des données de formation supplémentaires pour ces entités.

Lorsque vous créez un outil de reconnaissance d'entités personnalisé à l'aide de fichiers PDF annotés, vous pouvez l'utiliser avec différents formats de fichiers d'entrée : texte brut, fichiers image (JPG, PNG, TIFF), fichiers PDF et documents Word, sans aucun prétraitement ni aplatissement du document. Amazon Comprehend ne prend pas en charge l'annotation de fichiers image ou de documents Word.

> **Note**
>
> Un outil de reconnaissance d'entités personnalisé utilisant des fichiers PDF annotés ne prend en charge que les documents en anglais.

Après avoir créé un outil de reconnaissance d'entités personnalisé, vous pouvez suivre la progression de la demande à l'aide de l'DescribeEntityRecognizeropération. Une fois le `Status` champ définiTRAINED, le modèle de reconnaissance est prêt à être utilisé pour une reconnaissance d'entité personnalisée.

Rubriques

- Entraînez des outils de reconnaissance personnalisés (console)
- Formez des outils de reconnaissance d'entités personnalisés (API)
- Métriques de reconnaissance d'entités personnalisées

Entraînez des outils de reconnaissance personnalisés (console)

Vous pouvez créer des outils de reconnaissance d'entités personnalisés à l'aide de la console Amazon Comprehend. Cette section explique comment créer et entraîner un outil de reconnaissance d'entités personnalisé.

Création d'un outil de reconnaissance d'entités personnalisé à l'aide de la console - format CSV

Pour créer le système de reconnaissance d'entités personnalisé, fournissez d'abord un jeu de données pour entraîner votre modèle. Avec cet ensemble de données, incluez l'un des éléments suivants : un ensemble de documents annotés ou une liste d'entités et leur étiquette de type, ainsi qu'un ensemble de documents contenant ces entités. Pour plus d’informations, consultez Reconnaissance d'entités personnalisée.

Pour entraîner un outil de reconnaissance d'entités personnalisé à l'aide d'un fichier CSV

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Personnalisation, puis Reconnaissance d'entité personnalisée.
3. Choisissez Créer un nouveau modèle.
4. Donnez un nom au dispositif de reconnaissance. Le nom doit être unique au sein de la région et du compte.
5. Sélectionnez la langue.
6. Sous Type d'entité personnalisé, entrez une étiquette personnalisée que vous souhaitez que le dispositif de reconnaissance trouve dans le jeu de données.

Le type d'entité doit être en majuscules, et s'il est composé de plusieurs mots, séparez-les par un trait de soulignement.

7. Choisissez Ajouter un type.
8. Si vous souhaitez ajouter un type d'entité supplémentaire, saisissez-le, puis choisissez Ajouter un type. Si vous souhaitez supprimer l'un des types d'entités que vous avez ajoutés, choisissez Supprimer le type, puis choisissez le type d'entité à supprimer de la liste. Un maximum de 25 types d'entités peuvent être répertoriés.
9. Pour chiffrer votre tâche de formation, choisissez le chiffrement Recognizer, puis choisissez d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

   - Si vous utilisez une clé associée au compte actuel, choisissez l'ID de clé KMS dans le champ ID de clé.
   - Si vous utilisez une clé associée à un autre compte, dans le champ ARN de la clé KMS, entrez l'ARN de l'ID de la clé.

   > **Note**
   >
   > Pour plus d'informations sur la création et l'utilisation de clés KMS et le chiffrement associé, consultez AWS Key Management Service.

10. Sous Spécifications des données, choisissez le format de vos documents de formation :

    - Fichier CSV : fichier CSV qui complète vos documents de formation. Le fichier CSV contient des informations sur les entités personnalisées que votre modèle entraîné détectera. Le format requis du fichier varie selon que vous fournissez des annotations ou une liste d'entités.
    - Manifeste augmenté : ensemble de données étiqueté produit par Amazon SageMaker Ground Truth. Ce fichier est au format de lignes JSON. Chaque ligne est un objet JSON complet qui contient un document de formation et ses étiquettes. Chaque étiquette indique une entité nommée dans le document de formation. Vous pouvez fournir jusqu'à 5 fichiers manifestes augmentés.

    Pour plus d'informations sur les formats disponibles et pour des exemples, consultezFormer des modèles de reconnaissance d'entités personnalisés.

11. Sous Type de formation, choisissez le type de formation à utiliser :

- Utilisation des annotations et des documents de formation
- Utilisation de la liste des entités et des documents de formation

Si vous choisissez des annotations, entrez l'URL du fichier d'annotations dans Amazon S3. Vous pouvez également accéder au compartiment ou au dossier dans Amazon S3 où se trouvent les fichiers d'annotation et choisir Browse S3.

Si vous choisissez une liste d'entités, entrez l'URL de la liste d'entités dans Amazon S3. Vous pouvez également accéder au compartiment ou au dossier dans Amazon S3 où se trouve la liste des entités et choisir Browse S3.

12. Entrez l'URL d'un ensemble de données d'entrée contenant les documents de formation dans Amazon S3. Vous pouvez également accéder au compartiment ou au dossier d'Amazon S3 où se trouvent les documents de formation et choisir Sélectionner un dossier.
13. Sous Ensemble de données de test, sélectionnez la manière dont vous souhaitez évaluer les performances de votre modèle entraîné. Vous pouvez le faire à la fois pour les annotations et les types d'entraînement par liste d'entités.

    - Autosplit : Autosplit sélectionne automatiquement 10 % des données d'entraînement que vous avez fournies pour les utiliser comme données de test
    - (Facultatif) Fourni par le client : lorsque vous sélectionnez Fourni par le client, vous pouvez spécifier exactement les données de test que vous souhaitez utiliser.
14. Si vous sélectionnez l'ensemble de données de test fourni par le client, entrez l'URL du fichier d'annotations dans Amazon S3. Vous pouvez également accéder au compartiment ou au dossier dans Amazon S3 où se trouvent les fichiers d'annotation et choisir Sélectionner un dossier.
15. Dans la section Choisissez un rôle IAM, sélectionnez un rôle IAM existant ou créez-en un nouveau.

    - Choisissez un rôle IAM existant : sélectionnez cette option si vous possédez déjà un rôle IAM autorisé à accéder aux compartiments Amazon S3 en entrée et en sortie.
    - Créer un nouveau rôle IAM : sélectionnez cette option lorsque vous souhaitez créer un nouveau rôle IAM avec les autorisations appropriées permettant à Amazon Comprehend d'accéder aux compartiments d'entrée et de sortie.

> **Note**
>
> Si les documents d'entrée sont chiffrés, le rôle IAM utilisé doit disposer d'une `kms:Decrypt` autorisation. Pour de plus amples informations, veuillez consulter Autorisations requises pour utiliser le chiffrement KMS.

16. (Facultatif) Pour lancer vos ressources dans Amazon Comprehend à partir d'un VPC, entrez l'ID du VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseau (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.
    2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

    > **Note**
    >
    > Lorsque vous utilisez un VPC avec votre tâche de reconnaissance d'entité personnalisée, le VPC `DataAccessRole` utilisé pour les opérations de création et de démarrage doit disposer d'autorisations sur le VPC à partir duquel les documents d'entrée et le compartiment de sortie sont accessibles.

17. (Facultatif) Pour ajouter une balise au système de reconnaissance d'entités personnalisé, entrez une paire clé-valeur sous Balises. Choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le dispositif de reconnaissance, choisissez Supprimer le tag.
18. Choisissez Train.

Le nouveau dispositif de reconnaissance apparaîtra alors dans la liste et indiquera son statut. Il s'affichera d'abord sous la forme`Submitted`. Il s'affichera ensuite `Training` pour un classificateur qui traite des documents de formation, `Trained` pour un classificateur prêt à être utilisé et `In error` pour un classificateur comportant une erreur. Vous pouvez cliquer sur une tâche pour obtenir plus d'informations sur le système de reconnaissance, y compris les éventuels messages d'erreur.

Création d'un outil de reconnaissance d'entités personnalisé à l'aide de la console - manifeste augmenté

Pour entraîner un outil de reconnaissance d'entités personnalisé avec un document en texte brut, PDF ou Word

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Personnalisation, puis Reconnaissance d'entité personnalisée.
3. Choisissez Train recognizer.
4. Donnez un nom au dispositif de reconnaissance. Le nom doit être unique au sein de la région et du compte.
5. Sélectionnez la langue. Remarque : Si vous entraînez un document PDF ou Word, l'anglais est la langue prise en charge.
6. Sous Type d'entité personnalisé, entrez une étiquette personnalisée que vous souhaitez que le dispositif de reconnaissance trouve dans le jeu de données.

   Le type d'entité doit être en majuscules, et s'il est composé de plusieurs mots, séparez-les par un trait de soulignement.
7. Choisissez Ajouter un type.
8. Si vous souhaitez ajouter un type d'entité supplémentaire, saisissez-le, puis choisissez Ajouter un type. Si vous souhaitez supprimer l'un des types d'entités que vous avez ajoutés, choisissez Supprimer le type, puis choisissez le type d'entité à supprimer de la liste. Un maximum de 25 types d'entités peuvent être répertoriés.
9. Pour chiffrer votre tâche de formation, choisissez le chiffrement Recognizer, puis choisissez d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

   - Si vous utilisez une clé associée au compte actuel, choisissez l'ID de clé KMS dans le champ ID de clé.
   - Si vous utilisez une clé associée à un autre compte, dans le champ ARN de la clé KMS, entrez l'ARN de l'ID de la clé.

> **Note**
>
> Pour plus d'informations sur la création et l'utilisation de clés KMS et le chiffrement associé, consultez AWS Key Management Service.

10. Sous Données d'entraînement, choisissez le manifeste augmenté comme format de données :

    - Manifeste augmenté : ensemble de données étiqueté produit par Amazon SageMaker Ground Truth. Ce fichier est au format de lignes JSON. Chaque ligne du fichier est un objet JSON complet qui contient un document de formation et ses étiquettes. Chaque étiquette indique une entité nommée dans le document de formation. Vous pouvez fournir jusqu'à 5 fichiers manifestes augmentés. Si vous utilisez des documents PDF pour les données d'entraînement, vous devez sélectionner le manifeste augmenté. Vous pouvez fournir jusqu'à 5 fichiers manifestes augmentés. Pour chaque fichier, vous pouvez nommer jusqu'à 5 attributs à utiliser comme données d'entraînement.

    Pour plus d'informations sur les formats disponibles et pour des exemples, consultezFormer des modèles de reconnaissance d'entités personnalisés.

11. Sélectionnez le type de modèle d'entraînement.

    Si vous avez sélectionné Documents en texte brut, sous Emplacement d'entrée, entrez l'URL Amazon S3 du fichier manifeste augmenté Amazon SageMaker AIGround Truth. Vous pouvez également accéder au compartiment ou au dossier dans Amazon S3 où se trouvent les manifestes augmentés et choisir Sélectionner un dossier.

12. Sous Nom de l'attribut, entrez le nom de l'attribut qui contient vos annotations. Si le fichier contient des annotations provenant de plusieurs tâches d'étiquetage en chaîne, ajoutez un attribut pour chaque tâche. Dans ce cas, chaque attribut contient l'ensemble des annotations d'une tâche d'étiquetage. Remarque : Vous pouvez fournir jusqu'à 5 noms d'attributs pour chaque fichier.

13. Sélectionnez Ajouter.

14. Si vous avez sélectionné Documents PDF ou Word sous Emplacement de saisie, saisissez l'URL Amazon S3 du fichier manifeste augmenté Amazon SageMaker AI Ground Truth. Vous pouvez également accéder au compartiment ou au dossier dans Amazon S3 où se trouvent les manifestes augmentés et choisir Sélectionner un dossier.

15. Entrez le préfixe S3 pour vos fichiers de données d'annotation. Il s'agit des documents PDF que vous avez étiquetés.
16. Entrez le préfixe S3 pour vos documents source. Il s'agit des documents PDF originaux (objets de données) que vous avez fournis à Ground Truth pour votre travail d'étiquetage.
17. Entrez les noms des attributs contenant vos annotations. Remarque : Vous pouvez fournir jusqu'à 5 noms d'attributs pour chaque fichier. Tous les attributs de votre fichier que vous ne spécifiez pas sont ignorés.
18. Dans la section Rôle IAM, sélectionnez un rôle IAM existant ou créez-en un nouveau.

    - Choisissez un rôle IAM existant : sélectionnez cette option si vous possédez déjà un rôle IAM autorisé à accéder aux compartiments Amazon S3 en entrée et en sortie.
    - Créer un nouveau rôle IAM : sélectionnez cette option lorsque vous souhaitez créer un nouveau rôle IAM avec les autorisations appropriées permettant à Amazon Comprehend d'accéder aux compartiments d'entrée et de sortie.

    > **Note**
    >
    > Si les documents d'entrée sont chiffrés, le rôle IAM utilisé doit disposer d'une `kms:Decrypt` autorisation. Pour de plus amples informations, veuillez consulter Autorisations requises pour utiliser le chiffrement KMS.

19. (Facultatif) Pour lancer vos ressources dans Amazon Comprehend à partir d'un VPC, entrez l'ID du VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseau (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.
    2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

    > **Note**
    >
    > Lorsque vous utilisez un VPC avec votre tâche de reconnaissance d'entité personnalisée, le VPC `DataAccessRole` utilisé pour les opérations de création et

de démarrage doit disposer d'autorisations sur le VPC à partir duquel les documents d'entrée et le compartiment de sortie sont accessibles.

20. (Facultatif) Pour ajouter une balise au système de reconnaissance d'entités personnalisé, entrez une paire clé-valeur sous Balises. Choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le dispositif de reconnaissance, choisissez Supprimer le tag.
21. Choisissez Train.

Le nouveau dispositif de reconnaissance apparaîtra alors dans la liste et indiquera son statut. Il s'affichera d'abord sous la forme`Submitted`. Il s'affichera ensuite `Training` pour un classificateur qui traite des documents de formation, `Trained` pour un classificateur prêt à être utilisé et `In error` pour un classificateur comportant une erreur. Vous pouvez cliquer sur une tâche pour obtenir plus d'informations sur le système de reconnaissance, y compris les éventuels messages d'erreur.

Formez des outils de reconnaissance d'entités personnalisés (API)

Pour créer et entraîner un modèle de reconnaissance d'entité personnalisé, utilisez l'opération d'API Amazon Comprehend CreateEntityRecognizer

Rubriques

- Former des agents de reconnaissance d'entités personnalisés à l'aide du AWS Command Line Interface
- Former des agents de reconnaissance d'entités personnalisés à l'aide du AWS SDK pour Java
- Entraînement de dispositifs de reconnaissance d'entités personnalisés à l'aide de Python (Boto3)

Former des agents de reconnaissance d'entités personnalisés à l'aide du AWS Command Line Interface

Les exemples suivants illustrent l'utilisation de l'`CreateEntityRecognizer`opération et des autres opérations APIs associées au AWS CLI.

Les exemples sont formatés pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Créez un outil de reconnaissance d'entités personnalisé à l'aide de la commande `create-entity-recognizer` CLI. Pour plus d'informations sur le input-data-config paramètre, consultez le CreateEntityRecognizermanuel Amazon Comprehend API Reference.

```
aws comprehend create-entity-recognizer \
     --language-code en \
     --recognizer-name test-6 \
     --data-access-role-arn "arn:aws:iam::account number:role/service-role/
AmazonComprehendServiceRole-role" \
     --input-data-config "EntityTypes=[{Type=PERSON}],Documents={S3Uri=s3://Bucket
 Name/Bucket Path/documents},
                Annotations={S3Uri=s3://Bucket Name/Bucket Path/annotations}" \
     --region region
```

Répertoriez tous les dispositifs de reconnaissance d'entités d'une région à l'aide de la commande `list-entity-recognizers` CLI.

```
aws comprehend list-entity-recognizers \
     --region region
```

Vérifiez l'état des tâches des outils de reconnaissance d'entités personnalisés à l'aide de la commande `describe-entity-recognizer` CLI.

```
aws comprehend describe-entity-recognizer \
     --entity-recognizer-arn arn:aws:comprehend:region:account number:entity-
recognizer/test-6 \
     --region region
```

Former des agents de reconnaissance d'entités personnalisés à l'aide du AWS SDK pour Java

Cet exemple crée un outil de reconnaissance d'entités personnalisé et entraîne le modèle à l'aide de Java

Pour des exemples Amazon Comprehend utilisant Java, consultez les exemples Amazon Comprehend Java.

Entraînement de dispositifs de reconnaissance d'entités personnalisés à l'aide de Python (Boto3)

Instanciez le SDK Boto3 :

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Créez un outil de reconnaissance d'entités :

```
response = comprehend.create_entity_recognizer(
    RecognizerName="Recognizer-Name-Goes-Here-{}".format(str(uuid.uuid4())),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "EntityTypes": [
            {
                "Type": "ENTITY_TYPE"
            }
        ],
        "Documents": {
            "S3Uri": "s3://Bucket Name/Bucket Path/documents"
        },
        "Annotations": {
            "S3Uri": "s3://Bucket Name/Bucket Path/annotations"
        }
    }
)
recognizer_arn = response["EntityRecognizerArn"]
```

Répertoriez tous les outils de reconnaissance :

```
response = comprehend.list_entity_recognizers()
```

Attendez que le logiciel de reconnaissance atteigne le statut TRAINED :

```
while True:
    response = comprehend.describe_entity_recognizer(
        EntityRecognizerArn=recognizer_arn
    )
```

```
    status = response["EntityRecognizerProperties"]["Status"]
    if "IN_ERROR" == status:
        sys.exit(1)
    if "TRAINED" == status:
        break

    time.sleep(10)
```

Métriques de reconnaissance d'entités personnalisées

Amazon Comprehend vous fournit des indicateurs pour vous aider à estimer dans quelle mesure un outil de reconnaissance d'entités devrait fonctionner pour votre travail. Ils sont basés sur l'entraînement du modèle de reconnaissance. Ainsi, bien qu'ils représentent avec précision les performances du modèle pendant l'entraînement, ils ne constituent qu'une approximation des performances de l'API lors de la découverte des entités.

Les métriques sont renvoyées chaque fois que des métadonnées provenant d'un outil de reconnaissance d'entités entraîné sont renvoyées.

Amazon Comprehend prend en charge la formation d'un modèle sur un maximum de 25 entités à la fois. Lorsque des métriques sont renvoyées par un outil de reconnaissance d'entités entraîné, les scores sont calculés à la fois par rapport au dispositif de reconnaissance dans son ensemble (métriques globales) et pour chaque entité individuelle (métriques d'entité).

Trois mesures sont disponibles, à la fois sous forme de mesures globales et d'entités :

- Précision

  Cela indique la fraction d'entités produites par le système qui sont correctement identifiées et correctement étiquetées. Cela montre combien de fois l'identification de l'entité du modèle est vraiment une bonne identification. Il s'agit d'un pourcentage du nombre total d'identifications.

  En d'autres termes, la précision est basée sur les vrais positifs (tp) et les faux positifs (fp) et elle est calculée comme suit : precision = tp/(tp + fp).

  Par exemple, si un modèle prédit que deux exemples d'une entité sont présents dans un document, alors qu'il n'y en a qu'un, le résultat est un vrai positif et un faux positif. Dans ce cas, précision = 1/(1 + 1). La précision est de 50 %, car une entité est correcte sur les deux identifiées par le modèle.

- Rappel

  Cela indique le pourcentage d'entités présentes dans les documents qui sont correctement identifiées et étiquetées par le système. Mathématiquement, cela est défini en termes de nombre total d'identifications correctes, de vrais positifs (tp) et d'identifications manquées de faux négatifs (fn).

  Il est calculé comme recall = tp/(tp + fn). Par exemple, si un modèle identifie correctement une entité, mais omet deux autres instances où cette entité est présente, le résultat est un vrai positif et deux faux négatifs. Dans ce cas, rappelez = 1/(1 + 2). Le rappel est de 33,33 %, car une entité est correcte sur trois exemples possibles.

- Score de F1

  Il s'agit d'une combinaison des métriques Precision et Recall, qui mesure la précision globale du modèle pour la reconnaissance personnalisée des entités. Le score F1 est la moyenne harmonique des métriques Precision et Recall : F1 = 2 * Precision * Recall/(Precision + Recall).

  > **Note**
  >
  > Intuitivement, la moyenne harmonique pénalise les extrêmes plus que la simple moyenne ou d'autres moyens (exemple : `precision` = 0, `recall` = 1 pourrait être obtenu de manière triviale en prédisant tous les écarts possibles). Ici, la moyenne simple serait de 0,5, mais la F1 pénaliserait comme 0).

  Dans les exemples ci-dessus, `precision` = 50 % et `recall` = 33,33 %, donc F1 = 2 * 0,5 * 0,3333/(0,5 + 0,3333). Le score F1 est de 0,3975, soit 39,75 %.

Métriques globales et individuelles des entités

La relation entre les métriques globales et individuelles des entités peut être observée lors de l'analyse de la phrase suivante pour les entités qui sont soit un lieu, soit une personne.

```
John Washington and his friend Smith live in San Francisco, work in San Diego, and own
    a house in Seattle.
```

Dans notre exemple, le modèle fait les prédictions suivantes.

```
John Washington = Person
Smith = Place
San Francisco = Place
San Diego = Place
Seattle = Person
```

Cependant, les prévisions auraient dû être les suivantes.

```
John Washington = Person
Smith = Person
San Francisco = Place
San Diego = Place
Seattle = Place
```

Les mesures de chaque entité à cet effet seraient les suivantes :

```
entity:  Person
  True positive (TP) = 1 (because John Washington is correctly predicted to be a
    Person).
  False positive (FP) = 1 (because Seattle is incorrectly predicted to be a Person,
    but is actually a Place).
  False negative (FN) = 1 (because Smith is incorrectly predicted to be a Place, but
    is actually a Person).
  Precision = 1 / (1 + 1) = 0.5 or 50%
  Recall = 1 / (1+1) = 0.5 or 50%
  F1 Score = 2 * 0.5 * 0.5 / (0.5 + 0.5) = 0.5 or 50%

entity:  Place
  TP = 2 (because San Francisco and San Diego are each correctly predicted to be a
    Place).
  FP = 1 (because Smith is incorrectly predicted to be a Place, but is actually a
    Person).
  FN = 1 (because Seattle is incorrectly predicted to be a Person, but is actually a
    Place).
  Precision = 2 / (2+1) = 0.6667 or 66.67%
  Recall = 2 / (2+1) = 0.6667 or 66.67%
  F1 Score = 2 * 0.6667 * 0.6667 / (0.6667 + 0.6667) = 0.6667 or  66.67%
```

Les indicateurs globaux à cet égard seraient les suivants :

À l'échelle mondiale :

```
Global:
  TP = 3 (because John Washington, San Francisco and San Diego are predicted
 correctly.
    This is also the sum of all individual entity TP).
  FP = 2 (because Seattle is predicted as Person and Smith is predicted as Place. This
    is the sum of all individual entity FP).
  FN = 2 (because Seattle is predicted as Person and Smith is predicted as Place. This
    is the sum of all individual FN).
  Global Precision = 3 / (3+2) = 0.6 or 60%
    (Global Precision = Global TP / (Global TP + Global FP))
  Global Recall = 3 / (3+2) = 0.6 or 60%
    (Global Recall = Global TP / (Global TP + Global FN))
  Global F1Score = 2 * 0.6 * 0.6 / (0.6 + 0.6) = 0.6 or 60%
    (Global F1Score = 2 * Global Precision *  Global Recall / (Global Precision +
    Global Recall))
```

Améliorer les performances du système de reconnaissance d'entités personnalisé

Ces indicateurs donnent un aperçu de la précision avec laquelle le modèle formé fonctionnera lorsque vous l'utiliserez pour identifier des entités. Voici quelques options que vous pouvez utiliser pour améliorer vos indicateurs s'ils sont inférieurs à vos attentes :

1. Selon que vous utilisez Annotations ouListes d'entités (texte brut uniquement), assurez-vous de suivre les directives de la documentation correspondante pour améliorer la qualité des données. Si vous observez de meilleurs indicateurs après avoir amélioré vos données et réentraîné le modèle, vous pouvez continuer à itérer et à améliorer la qualité des données afin d'améliorer les performances du modèle.
2. Si vous utilisez une liste d'entités, pensez plutôt à utiliser des annotations. Les annotations manuelles peuvent souvent améliorer vos résultats.
3. Si vous êtes certain qu'il n'y a aucun problème de qualité des données, mais que les indicateurs restent déraisonnablement bas, veuillez envoyer une demande d'assistance.

Exécution d'une analyse de reconnaissance personnalisée en temps réel

L'analyse en temps réel est utile pour les applications qui traitent de petits documents à leur arrivée. Par exemple, vous pouvez détecter des entités personnalisées dans les publications sur les réseaux sociaux, les tickets d'assistance ou les avis clients.

> **Avant de commencer**
>
> Vous avez besoin d'un modèle de reconnaissance d'entités personnalisé (également appelé outil de reconnaissance) avant de pouvoir détecter des entités personnalisées. Pour plus d'informations sur ces modèles, consultezthe section called “Modèles de reconnaissance d'entraînement”.

Un outil de reconnaissance entraîné avec des annotations en texte brut prend en charge la détection d'entités uniquement pour les documents en texte brut. Un outil de reconnaissance entraîné avec des annotations de documents PDF permet de détecter les entités pour les documents en texte brut, les images, les fichiers PDF et les documents Word. Pour plus d'informations sur les fichiers d'entrée, consultezEntrées pour une analyse personnalisée en temps réel.

Si vous prévoyez d'analyser des fichiers image ou des documents PDF numérisés, votre politique IAM doit autoriser l'utilisation de deux méthodes d'API Amazon Textract DetectDocumentText (AnalyzeDocumentet). Amazon Comprehend invoque ces méthodes lors de l'extraction de texte. Pour un exemple de politique, veuillez consulter Autorisations requises pour effectuer des actions d'analyse de documents.

Rubriques

- Analyse en temps réel pour la reconnaissance d'entités personnalisées (console)
- Analyse en temps réel pour la reconnaissance d'entités personnalisées (API)
- Sorties pour une analyse en temps réel

Analyse en temps réel pour la reconnaissance d'entités personnalisées (console)

Vous pouvez utiliser la console Amazon Comprehend pour exécuter une analyse en temps réel avec un modèle personnalisé. Tout d'abord, vous créez un point de terminaison pour exécuter l'analyse en temps réel. Après avoir créé le point de terminaison, vous exécutez l'analyse en temps réel.

Pour plus d'informations sur le provisionnement du débit des terminaux et les coûts associés, consultez. Utilisation des points de terminaison Amazon Comprehend

Rubriques

- Création d'un point de terminaison pour la détection d'entités personnalisées
- Exécution de la détection d'entités personnalisées en temps réel

Création d'un point de terminaison pour la détection d'entités personnalisées

Pour créer un point de terminaison (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Endpoints, puis cliquez sur le bouton Créer un point de terminaison. L'écran Créer un point de terminaison s'ouvre.
3. Donnez un nom au point de terminaison. Le nom doit être unique dans la région et le compte actuels.
4. Choisissez un modèle personnalisé auquel vous souhaitez associer le nouveau point de terminaison. Dans le menu déroulant, vous pouvez effectuer une recherche par nom de modèle.

   > **Note**
   >
   > Vous devez créer un modèle avant de pouvoir y associer un point de terminaison. Si vous n'avez pas encore de modèle, consultezFormer des modèles de reconnaissance d'entités personnalisés.

5. (Facultatif) Pour ajouter une balise au point de terminaison, entrez une paire clé-valeur sous Tags et choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le point de terminaison, choisissez Supprimer le tag.

6. Entrez le nombre d'unités d'inférence (IUs) à attribuer au point de terminaison. Chaque unité représente un débit de 100 caractères par seconde pour un maximum de deux documents par seconde. Pour plus d'informations sur le débit des terminaux, consultezUtilisation des points de terminaison Amazon Comprehend.
7. (Facultatif) Si vous créez un nouveau point de terminaison, vous avez la possibilité d'utiliser l'estimateur IU. L'estimateur peut vous aider à déterminer le nombre IUs de demandes. Le nombre d'unités d'inférence dépend du débit ou du nombre de caractères que vous souhaitez analyser par seconde.
8. À partir du récapitulatif des achats, passez en revue le coût horaire, quotidien et mensuel estimé de votre terminal.
9. Cochez cette case si vous comprenez que votre compte est débité pour le terminal à partir du moment où il démarre et jusqu'à ce que vous le supprimiez.
10. Choisissez Créer un point de terminaison.

Exécution de la détection d'entités personnalisées en temps réel

Après avoir créé un point de terminaison pour votre modèle de reconnaissance d'entités personnalisé, vous pouvez exécuter une analyse en temps réel pour détecter les entités dans des documents individuels.

Procédez comme suit pour détecter les entités personnalisées dans votre texte à l'aide de la console Amazon Comprehend.

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, sélectionnez Analyse en temps réel.
3. Dans la section Texte d'entrée, pour Type d'analyse, sélectionnez Personnalisé.
4. Pour Sélectionner un point de terminaison, choisissez le point de terminaison associé au modèle de détection d'entités que vous souhaitez utiliser.
5. Pour spécifier les données d'entrée à analyser, vous pouvez saisir du texte ou télécharger un fichier.
   - Pour saisir du texte :
     a. Choisissez Saisie de texte.
     b. Entrez le texte que vous souhaitez analyser.

- Pour télécharger un fichier, procédez comme suit :

  a. Choisissez Charger le fichier et entrez le nom du fichier à télécharger.

  b. (Facultatif) Sous Actions de lecture avancées, vous pouvez remplacer les actions par défaut pour l'extraction de texte. Pour plus de détails, consultez Configuration des options d'extraction de texte.

6. Choisissez Analyser. La console affiche le résultat de l'analyse, ainsi qu'une évaluation de confiance.

Analyse en temps réel pour la reconnaissance d'entités personnalisées (API)

Vous pouvez utiliser l'API Amazon Comprehend pour exécuter une analyse en temps réel avec un modèle personnalisé. Tout d'abord, vous créez un point de terminaison pour exécuter l'analyse en temps réel. Après avoir créé le point de terminaison, vous exécutez l'analyse en temps réel.

Pour plus d'informations sur le provisionnement du débit des terminaux et les coûts associés, consultez. Utilisation des points de terminaison Amazon Comprehend

Rubriques

- Création d'un point de terminaison pour la détection d'entités personnalisées
- Exécution de la détection d'entités personnalisées en temps réel

Création d'un point de terminaison pour la détection d'entités personnalisées

Pour plus d'informations sur les coûts associés aux points de terminaison, consultezUtilisation des points de terminaison Amazon Comprehend.

Création d'un point de terminaison avec AWS CLI

Pour créer un point de terminaison à l'aide de AWS CLI, utilisez la `create-endpoint` commande :

```
$ aws comprehend create-endpoint \
> --desired-inference-units number of inference units \
> --endpoint-name endpoint name \
> --model-arn arn:aws:comprehend:region:account-id:model/example \
> --tags Key=Key,Value=Value
```

Si votre commande aboutit, Amazon Comprehend répond avec l'ARN du point de terminaison :

```
{
   "EndpointArn": "Arn"
}
```

Pour plus d'informations sur cette commande, ses arguments de paramètres et sa sortie, consultez `create-endpoint`la référence des AWS CLI commandes.

Exécution de la détection d'entités personnalisées en temps réel

Après avoir créé un point de terminaison pour votre modèle de reconnaissance d'entité personnalisé, vous utilisez le point de terminaison pour exécuter l'opération DetectEntitiesd'API. Vous pouvez saisir du texte à l'aide du `bytes` paramètre `text` or. Entrez les autres types d'entrée à l'aide du `bytes` paramètre.

Pour les fichiers image et les fichiers PDF, vous pouvez utiliser le `DocumentReaderConfig` paramètre pour remplacer les actions d'extraction de texte par défaut. Pour plus de détails, consultez Configuration des options d'extraction de texte.

Détection d'entités dans le texte à l'aide du AWS CLI

Pour détecter des entités personnalisées dans le texte, exécutez la `detect-entities` commande avec le texte saisi dans le `text` paramètre.

Example : utilisez la CLI pour détecter les entités dans le texte saisi

```
$ aws comprehend detect-entities \
> --endpoint-arn arn \
> --language-code en \
> --text  "Andy Jassy is the CEO of Amazon."
```

Si votre commande aboutit, Amazon Comprehend répond par l'analyse. Pour chaque entité détectée par Amazon Comprehend, celui-ci fournit le type d'entité, le texte, l'emplacement et le score de confiance.

Détection d'entités dans des documents semi-structurés à l'aide du AWS CLI

Pour détecter des entités personnalisées dans un fichier PDF, Word ou image, exécutez la `detect-entities` commande avec le fichier d'entrée dans le `bytes` paramètre.

Example : utilisez la CLI pour détecter les entités dans un fichier image

Cet exemple montre comment transmettre le fichier image à l'aide de l'`file`boption permettant de coder les octets de l'image en base64. Pour plus d'informations, consultez la section Objets binaires volumineux dans le guide de AWS Command Line Interface l'utilisateur.

Cet exemple transmet également un fichier JSON nommé `config.json` pour définir les options d'extraction de texte.

```
$ aws comprehend detect-entities \
> --endpoint-arn arn \
> --language-code en \
> --bytes fileb://image1.jpg   \
> --document-reader-config file://config.json
```

Le fichier config.json contient le contenu suivant.

```
{
   "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
   "DocumentReadAction": "TEXTRACT_DETECT_DOCUMENT_TEXT"
}
```

Pour plus d'informations sur la syntaxe des commandes, consultez DetectEntitiesle manuel Amazon Comprehend API Reference.

Sorties pour une analyse en temps réel

Sorties pour entrées de texte

Si vous saisissez du texte à l'aide du `Text` paramètre, la sortie consiste en un tableau d'entités détectées par l'analyse. L'exemple suivant montre une analyse qui a détecté deux entités JUDGE.

```
{
        "Entities":
        [
            {
                "BeginOffset": 0,
                "EndOffset": 22,
                "Score": 0.9763959646224976,
```

```
                "Text": "John Johnson",
                "Type": "JUDGE"
            },
            {
                "BeginOffset": 11,
                "EndOffset": 15,
                "Score": 0.9615424871444702,
                "Text": "Thomas Kincaid",
                "Type": "JUDGE"
            }
        ]
    }
```

Sorties pour entrées semi-structurées

Pour un document d'entrée semi-structuré ou un fichier texte, la sortie peut inclure les champs supplémentaires suivants :

- DocumentMetadata — Informations d'extraction relatives au document. Les métadonnées incluent une liste de pages du document, avec le nombre de caractères extraits de chaque page. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- DocumentType — Type de document pour chaque page du document d'entrée. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre.
- Blocs : informations relatives à chaque bloc de texte du document d'entrée. Les blocs sont imbriqués. Un bloc de page contient un bloc pour chaque ligne de texte, qui contient un bloc pour chaque mot. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre.
- BlockReferences — Une référence à chaque bloc pour cette entité. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre. Le champ n'est pas présent pour les fichiers texte.
- Erreurs : erreurs au niveau de la page détectées par le système lors du traitement du document d'entrée. Le champ est vide si le système n'a détecté aucune erreur.

Pour obtenir une description de ces champs de sortie, consultez le DetectEntitiesmanuel Amazon Comprehend API Reference. Pour plus d'informations sur les éléments de mise en page, consultez les objets d'analyse Amazon Textract dans le manuel Amazon Textract Developer Guide.

L'exemple suivant montre le résultat d'un document d'entrée PDF numérisé d'une page.

```
{
```

```
    "Entities": [{
        "Score": 0.9984670877456665,
        "Type": "DATE-TIME",
        "Text": "September 4,",
        "BlockReferences": [{
            "BlockId": "42dcaaee-c484-4b5d-9e3f-ae0be928b3e1",
            "BeginOffset": 0,
            "EndOffset": 12,
            "ChildBlocks": [{
                    "ChildBlockId": "6e9cbb43-f8be-4da0-9a4b-ff9a6c350a14",
                    "BeginOffset": 0,
                    "EndOffset": 9
                },
                {
                    "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
                    "BeginOffset": 0,
                    "EndOffset": 2
                },
                {
                    "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
                    "BeginOffset": 0,
                    "EndOffset": 2
                }
            ]
        }]
    }],
    "DocumentMetadata": {
        "Pages": 1,
        "ExtractedCharacters": [{
            "Page": 1,
            "Count": 609
        }]
    },
    "DocumentType": [{
        "Page": 1,
        "Type": "SCANNED_PDF"
    }],
    "Blocks": [{
        "Id": "ee82edf3-28de-4d63-8883-40e2e4938ccb",
        "BlockType": "LINE",
        "Text": "Your Band",
        "Page": 1,
        "Geometry": {
            "BoundingBox": {
```

```
                "Height": 0.024125460535287857,
                "Left": 0.11745482683181763,
                "Top": 0.06821706146001816,
                "Width": 0.12074867635965347
            },
            "Polygon": [{
                    "X": 0.11745482683181763,
                    "Y": 0.06821706146001816
                },
                {
                    "X": 0.2382034957408905,
                    "Y": 0.06821706146001816
                },
                {
                    "X": 0.2382034957408905,
                    "Y": 0.09234252572059631
                },
                {
                    "X": 0.11745482683181763,
                    "Y": 0.09234252572059631
                }
            ]
        },
        "Relationships": [{
            "Ids": [
                "b105c561-c8d9-485a-a728-7a5b1a308935",
                "60ecb119-3173-4de2-8c5d-de182a5f86a5"
            ],
            "Type": "CHILD"
        }]
    }]
}
```

L'exemple suivant montre le résultat pour l'analyse d'un document PDF natif.

Example Exemple de résultat d'une analyse de reconnaissance d'entités personnalisée d'un document PDF

```
{
        "Blocks":
        [
            {
                "BlockType": "LINE",
```

```
                "Geometry":
                {
                    "BoundingBox":
                    {
                        "Height": 0.012575757575757575,
                        "Left": 0.0,
                        "Top": 0.0015063131313131314,
                        "Width": 0.02262091503267974
                    },
                    "Polygon":
                    [
                        {
                            "X": 0.0,
                            "Y": 0.0015063131313131314
                        },
                        {
                            "X": 0.02262091503267974,
                            "Y": 0.0015063131313131314
                        },
                        {
                            "X": 0.02262091503267974,
                            "Y": 0.014082070707070706
                        },
                        {
                            "X": 0.0,
                            "Y": 0.014082070707070706
                        }
                    ]
                },
                "Id": "4330efed-6334-4fc4-ba48-e050afa95c8d",
                "Page": 1,
                "Relationships":
                [
                    {
                        "ids":
                        [
                            "f343ce48-583d-4abe-b84b-a232e266450f"
                        ],
                        "type": "CHILD"
                    }
                ],
                "Text": "S-3"
            },
            {
```

```
                "BlockType": "WORD",
                "Geometry":
                {
                    "BoundingBox":
                    {
                        "Height": 0.012575757575757575,
                        "Left": 0.0,
                        "Top": 0.0015063131313131314,
                        "Width": 0.02262091503267974
                    },
                    "Polygon":
                    [
                        {
                            "X": 0.0,
                            "Y": 0.0015063131313131314
                        },
                        {
                            "X": 0.02262091503267974,
                            "Y": 0.0015063131313131314
                        },
                        {
                            "X": 0.02262091503267974,
                            "Y": 0.014082070707070706
                        },
                        {
                            "X": 0.0,
                            "Y": 0.014082070707070706
                        }
                    ]
                },
                "Id": "f343ce48-583d-4abe-b84b-a232e266450f",
                "Page": 1,
                "Relationships":
                [],
                "Text": "S-3"
            }
        ],
        "DocumentMetadata":
        {
            "PageNumber": 1,
            "Pages": 1
        },
        "DocumentType": "NativePDF",
        "Entities":
```

```
        [
            {
                "BlockReferences":
                [
                    {
                        "BeginOffset": 25,
                        "BlockId": "4330efed-6334-4fc4-ba48-e050afa95c8d",
                        "ChildBlocks":
                        [
                            {
                                "BeginOffset": 1,
                                "ChildBlockId": "cbba5534-ac69-4bc4-beef-306c659f70a6",
                                "EndOffset": 6
                            }
                        ],
                        "EndOffset": 30
                    }
                ],
                "Score": 0.9998825926329088,
                "Text": "0.001",
                "Type": "OFFERING_PRICE"
            },
            {
                "BlockReferences":
                [
                    {
                        "BeginOffset": 41,
                        "BlockId": "f343ce48-583d-4abe-b84b-a232e266450f",
                        "ChildBlocks":
                        [
                            {
                                "BeginOffset": 0,
                                "ChildBlockId": "292a2e26-21f0-401b-a2bf-03aa4c47f787",
                                "EndOffset": 9
                            }
                        ],
                        "EndOffset": 50
                    }
                ],
                "Score": 0.9809727537330395,
                "Text": "6,097,560",
                "Type": "OFFERED_SHARES"
            }
        ],
```

```
        "File": "example.pdf",
        "Version": "2021-04-30"
    }
```

Exécution de tâches d'analyse pour une reconnaissance d'entités personnalisée

Vous pouvez exécuter une tâche d'analyse asynchrone pour détecter des entités personnalisées dans un ensemble d'un ou de plusieurs documents.

> **Avant de commencer**
>
> Vous avez besoin d'un modèle de reconnaissance d'entités personnalisé (également appelé outil de reconnaissance) avant de pouvoir détecter des entités personnalisées. Pour plus d'informations sur ces modèles, consultezthe section called “Modèles de reconnaissance d'entraînement”.

Un outil de reconnaissance entraîné avec des annotations en texte brut prend en charge la détection d'entités uniquement pour les documents en texte brut. Un outil de reconnaissance formé à l'aide d'annotations de documents PDF permet de détecter les entités pour les documents en texte brut, les images, les fichiers PDF et les documents Word. Pour les fichiers autres que des fichiers texte, Amazon Comprehend effectue une extraction de texte avant d'exécuter l'analyse. Pour plus d'informations sur les fichiers d'entrée, consultezEntrées pour une analyse personnalisée asynchrone.

Si vous prévoyez d'analyser des fichiers image ou des documents PDF numérisés, votre politique IAM doit autoriser l'utilisation de deux méthodes d'API Amazon Textract DetectDocumentText (AnalyzeDocumentet). Amazon Comprehend invoque ces méthodes lors de l'extraction de texte. Pour un exemple de politique, consultez Autorisations requises pour effectuer des actions d'analyse de documents.

Pour exécuter une tâche d'analyse asynchrone, vous devez suivre les étapes générales suivantes :

1. Stockez les documents dans un compartiment Amazon S3.
2. Utilisez l'API ou la console pour démarrer le travail d'analyse.
3. Surveillez la progression de la tâche d'analyse.

4. Une fois le travail terminé, récupérez les résultats de l'analyse dans le compartiment S3 que vous avez spécifié lorsque vous avez démarré le travail.

Rubriques

- Démarrage d'une tâche de détection d'entités personnalisée (console)
- Démarrage d'une tâche de détection d'entités personnalisée (API)
- Sorties pour les tâches d'analyse asynchrones

Démarrage d'une tâche de détection d'entités personnalisée (console)

Vous pouvez utiliser la console pour démarrer et surveiller une tâche d'analyse asynchrone pour une reconnaissance d'entité personnalisée.

Pour démarrer une tâche d'analyse asynchrone

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Tâches d'analyse, puis sélectionnez Créer une tâche.
3. Donnez un nom à la tâche de classification. Le nom doit être unique pour votre compte et pour votre région actuelle.
4. Sous Type d'analyse, sélectionnez Reconnaissance d'entité personnalisée.
5. Dans le modèle de reconnaissance, choisissez le système de reconnaissance d'entités personnalisé à utiliser.
6. Dans Version, choisissez la version de reconnaissance à utiliser.
7. (Facultatif) Si vous choisissez de chiffrer les données utilisées par Amazon Comprehend lors du traitement de votre tâche, choisissez Job encryption. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.
   - Si vous utilisez une clé associée au compte actuel, choisissez l'ID de clé pour l'ID de clé KMS.
   - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'ID de clé sous ARN de la clé KMS.

> **Note**
>
> Pour plus d'informations sur la création et l'utilisation des clés KMS et le chiffrement associé, consultez la section Service de gestion des clés (KMS).

8. Sous Données d'entrée, entrez l'emplacement du compartiment Amazon S3 qui contient vos documents d'entrée ou naviguez jusqu'à celui-ci en choisissant Browse S3. Ce compartiment doit se trouver dans la même région que l'API que vous appelez. Le rôle IAM que vous utilisez pour les autorisations d'accès pour la tâche d'analyse doit disposer d'autorisations de lecture pour le compartiment S3.
9. (Facultatif) Pour le format d'entrée, vous pouvez choisir le format des documents d'entrée. Le format peut être un document par fichier ou un document par ligne dans un seul fichier. Un document par ligne s'applique uniquement aux documents texte.
10. (Facultatif) En mode lecture de document, vous pouvez remplacer les actions d'extraction de texte par défaut. Pour de plus amples informations, veuillez consulter Configuration des options d'extraction de texte.
11. Sous Données de sortie, entrez l'emplacement du compartiment Amazon S3 dans lequel Amazon Comprehend doit écrire les données de sortie de la tâche ou accédez à celui-ci en choisissant Browse S3. Ce compartiment doit se trouver dans la même région que l'API que vous appelez. Le rôle IAM que vous utilisez pour les autorisations d'accès pour la tâche de classification doit disposer d'autorisations d'écriture pour le compartiment S3.
12. (Facultatif) Si vous choisissez de chiffrer le résultat de sortie de votre tâche, choisissez Chiffrement. Choisissez ensuite d'utiliser une clé KMS associée au compte actuel ou une clé provenant d'un autre compte.

    - Si vous utilisez une clé associée au compte actuel, choisissez l'alias de clé ou l'ID pour l'ID de clé KMS.
    - Si vous utilisez une clé associée à un autre compte, entrez l'ARN de l'alias ou de l'ID de clé sous ID de clé KMS.
13. (Facultatif) Pour lancer vos ressources dans Amazon Comprehend à partir d'un VPC, entrez l'ID du VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseau (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.

2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

> **Note**
>
> Lorsque vous utilisez un VPC avec votre tâche d'analyse, le VPC `DataAccessRole` utilisé pour les opérations de création et de démarrage doit disposer des autorisations d'accès au VPC qui accède au compartiment de sortie.

14. Choisissez Create job pour créer le job de reconnaissance d'entité.

Démarrage d'une tâche de détection d'entités personnalisée (API)

Vous pouvez utiliser l'API pour démarrer et surveiller une tâche d'analyse asynchrone pour une reconnaissance d'entité personnalisée.

Pour démarrer une tâche de détection d'entité personnalisée avec l'StartEntitiesDetectionJobopération, vous devez fournir le EntityRecognizerArn, qui est l'Amazon Resource Name (ARN) du modèle entraîné. Vous pouvez trouver cet ARN dans la réponse à l'CreateEntityRecognizeropération.

Rubriques

- Détection d'entités personnalisées à l'aide du AWS Command Line Interface
- Détection d'entités personnalisées à l'aide du AWS SDK pour Java
- Détection d'entités personnalisées à l'aide du AWS SDK pour Python (Boto3)
- Remplacer les actions de l'API pour les fichiers PDF

Détection d'entités personnalisées à l'aide du AWS Command Line Interface

Utilisez l'exemple suivant pour les environnements Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^). Pour détecter des entités personnalisées dans un ensemble de documents, utilisez la syntaxe de demande suivante :

```
aws comprehend start-entities-detection-job \
```

```
     --entity-recognizer-arn "arn:aws:comprehend:region:account number:entity-
recognizer/test-6" \
     --job-name infer-1 \
     --data-access-role-arn "arn:aws:iam::account number:role/service-role/
AmazonComprehendServiceRole-role" \
     --language-code en \
     --input-data-config "S3Uri=s3://Bucket Name/Bucket Path" \
     --output-data-config "S3Uri=s3://Bucket Name/Bucket Path/" \
     --region region
```

Amazon Comprehend répond par le `JobID` `JobStatus` et renvoie le résultat de la tâche dans le compartiment S3 que vous avez spécifié dans la demande.

Détection d'entités personnalisées à l'aide du AWS SDK pour Java

Pour des exemples Amazon Comprehend utilisant Java, consultez les exemples Amazon Comprehend Java.

Détection d'entités personnalisées à l'aide du AWS SDK pour Python (Boto3)

Cet exemple crée un outil de reconnaissance d'entités personnalisé, entraîne le modèle, puis l'exécute dans une tâche de reconnaissance d'entités à l'aide du. AWS SDK pour Python (Boto3)

Instanciez le SDK pour Python.

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Créez un outil de reconnaissance d'entités :

```
response = comprehend.create_entity_recognizer(
    RecognizerName="Recognizer-Name-Goes-Here-{}".format(str(uuid.uuid4())),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "EntityTypes": [
            {
                "Type": "ENTITY_TYPE"
            }
        ],
        "Documents": {
            "S3Uri": "s3://Bucket Name/Bucket Path/documents"
```

```
        },
        "Annotations": {
            "S3Uri": "s3://Bucket Name/Bucket Path/annotations"
        }
    }
)
recognizer_arn = response["EntityRecognizerArn"]
```

Répertoriez tous les outils de reconnaissance :

```
response = comprehend.list_entity_recognizers()
```

Attendez que le système de reconnaissance d'entités atteigne le statut TRAINED :

```
while True:
    response = comprehend.describe_entity_recognizer(
        EntityRecognizerArn=recognizer_arn
    )

    status = response["EntityRecognizerProperties"]["Status"]
    if "IN_ERROR" == status:
        sys.exit(1)
    if "TRAINED" == status:
        break

    time.sleep(10)
```

Démarrez une tâche de détection d'entités personnalisée :

```
response = comprehend.start_entities_detection_job(
    EntityRecognizerArn=recognizer_arn,
    JobName="Detection-Job-Name-{}".format(str(uuid.uuid4())),
    LanguageCode="en",
    DataAccessRoleArn="Role ARN",
    InputDataConfig={
        "InputFormat": "ONE_DOC_PER_LINE",
        "S3Uri": "s3://Bucket Name/Bucket Path/documents"
    },
    OutputDataConfig={
        "S3Uri": "s3://Bucket Name/Bucket Path/output"
    }
)
```

Remplacer les actions de l'API pour les fichiers PDF

Pour les fichiers image et les fichiers PDF, vous pouvez remplacer les actions d'extraction par défaut à l'aide du `DocumentReaderConfig` paramètre in`InputDataConfig`.

L'exemple suivant définit un fichier JSON nommé myInputData Config.json pour définir les `InputDataConfig` valeurs. Il est configuré `DocumentReadConfig` pour utiliser l'`DetectDocumentText`API Amazon Textract pour tous les fichiers PDF.

Example

```
"InputDataConfig": {
  "S3Uri": s3://Bucket Name/Bucket Path",
  "InputFormat": "ONE_DOC_PER_FILE",
  "DocumentReaderConfig": {
      "DocumentReadAction": "TEXTRACT_DETECT_DOCUMENT_TEXT",
      "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION"
  }
}
```

Dans l'`StartEntitiesDetectionJob`opération, spécifiez le fichier myInputData Config.json comme paramètre : `InputDataConfig`

```
  --input-data-config file://myInputDataConfig.json
```

Pour plus d'informations sur les `DocumentReaderConfig` paramètres, consultezConfiguration des options d'extraction de texte.

Sorties pour les tâches d'analyse asynchrones

Une fois le travail d'analyse terminé, il stocke les résultats dans le compartiment S3 que vous avez spécifié dans la demande.

Sorties pour entrées de texte

Pour les fichiers de saisie de texte, la sortie consiste en une liste d'entités pour chaque document d'entrée.

L'exemple suivant montre la sortie de deux documents à partir d'un fichier d'entrée nommé50_docs, en utilisant un format de document par ligne.

```
{
        "File": "50_docs",
        "Line": 0,
        "Entities":
        [
            {
                "BeginOffset": 0,
                "EndOffset": 22,
                "Score": 0.9763959646224976,
                "Text": "John Johnson",
                "Type": "JUDGE"
            }
        ]
    }
    {
        "File": "50_docs",
        "Line": 1,
        "Entities":
        [
            {
                "BeginOffset": 11,
                "EndOffset": 15,
                "Score": 0.9615424871444702,
                "Text": "Thomas Kincaid",
                "Type": "JUDGE"
            }
        ]
    }
```


Sorties pour entrées semi-structurées

Pour les documents d'entrée semi-structurés, la sortie peut inclure les champs supplémentaires suivants :

- DocumentMetadata — Informations d'extraction relatives au document. Les métadonnées incluent une liste de pages du document, avec le nombre de caractères extraits de chaque page. Ce champ est présent dans la réponse si la demande inclut le `Byte` paramètre.
- DocumentType — Type de document pour chaque page du document d'entrée. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre.
- Blocs : informations relatives à chaque bloc de texte du document d'entrée. Les blocs peuvent s'imbriquer dans un bloc. Un bloc de page contient un bloc pour chaque ligne de texte, qui contient

un bloc pour chaque mot. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre.

- BlockReferences — Une référence à chaque bloc pour cette entité. Ce champ est présent dans la réponse à une demande incluant le `Byte` paramètre. Le champ n'est pas présent pour les fichiers texte.
- Erreurs : erreurs au niveau de la page détectées par le système lors du traitement du document d'entrée. Le champ est vide si le système n'a détecté aucune erreur.

Pour plus de détails sur ces champs de sortie, consultez DetectEntitiesle manuel Amazon Comprehend API Reference

L'exemple suivant montre le résultat d'un document d'entrée PDF natif d'une page.

Example Exemple de résultat d'une analyse de reconnaissance d'entités personnalisée d'un document PDF


```
{
        "Blocks":
        [
            {
                "BlockType": "LINE",
                "Geometry":
                {
                    "BoundingBox":
                    {
                        "Height": 0.012575757575757575,
                        "Left": 0.0,
                        "Top": 0.0015063131313131314,
                        "Width": 0.02262091503267974
                    },
                    "Polygon":
                    [
                        {
                            "X": 0.0,
                            "Y": 0.0015063131313131314
                        },
                        {
                            "X": 0.02262091503267974,
                            "Y": 0.0015063131313131314
                        },
                        {
```

```
                        "X": 0.02262091503267974,
                        "Y": 0.014082070707070706
                    },
                    {
                        "X": 0.0,
                        "Y": 0.014082070707070706
                    }
                ]
            },
            "Id": "4330efed-6334-4fc4-ba48-e050afa95c8d",
            "Page": 1,
            "Relationships":
            [
                {
                    "ids":
                    [
                        "f343ce48-583d-4abe-b84b-a232e266450f"
                    ],
                    "type": "CHILD"
                }
            ],
            "Text": "S-3"
        },
        {
            "BlockType": "WORD",
            "Geometry":
            {
                "BoundingBox":
                {
                    "Height": 0.012575757575757575,
                    "Left": 0.0,
                    "Top": 0.0015063131313131314,
                    "Width": 0.02262091503267974
                },
                "Polygon":
                [
                    {
                        "X": 0.0,
                        "Y": 0.0015063131313131314
                    },
                    {
                        "X": 0.02262091503267974,
                        "Y": 0.0015063131313131314
                    },
```

```
                    {
                        "X": 0.02262091503267974,
                        "Y": 0.014082070707070706
                    },
                    {
                        "X": 0.0,
                        "Y": 0.014082070707070706
                    }
                ]
            },
            "Id": "f343ce48-583d-4abe-b84b-a232e266450f",
            "Page": 1,
            "Relationships":
            [],
            "Text": "S-3"
        }
    ],
    "DocumentMetadata":
    {
        "PageNumber": 1,
        "Pages": 1
    },
    "DocumentType": "NativePDF",
    "Entities":
    [
        {
            "BlockReferences":
            [
                {
                    "BeginOffset": 25,
                    "BlockId": "4330efed-6334-4fc4-ba48-e050afa95c8d",
                    "ChildBlocks":
                    [
                        {
                            "BeginOffset": 1,
                            "ChildBlockId": "cbba5534-ac69-4bc4-beef-306c659f70a6",
                            "EndOffset": 6
                        }
                    ],
                    "EndOffset": 30
                }
            ],
            "Score": 0.9998825926329088,
            "Text": "0.001",
```

```
                "Type": "OFFERING_PRICE"
            },
            {
                "BlockReferences":
                [
                    {
                        "BeginOffset": 41,
                        "BlockId": "f343ce48-583d-4abe-b84b-a232e266450f",
                        "ChildBlocks":
                        [
                            {
                                "BeginOffset": 0,
                                "ChildBlockId": "292a2e26-21f0-401b-a2bf-03aa4c47f787",
                                "EndOffset": 9
                            }
                        ],
                        "EndOffset": 50
                    }
                ],
                "Score": 0.9809727537330395,
                "Text": "6,097,560",
                "Type": "OFFERED_SHARES"
            }
        ],
        "File": "example.pdf",
        "Version": "2021-04-30"
    }
```

Création et gestion de modèles personnalisés

Amazon Comprehend inclut des modèles NLP (traitement du langage naturel) intégrés que vous pouvez utiliser pour analyser des informations ou modéliser des sujets. Vous pouvez également utiliser Amazon Comprehend pour créer des modèles personnalisés pour la reconnaissance d'entités et la classification de documents.

Vous pouvez utiliser le versionnement des modèles pour suivre l'historique de votre modèle. Lorsque vous créez et entraînez une nouvelle version du modèle, vous pouvez apporter des modifications au jeu de données d'entraînement. Amazon Comprehend affiche les détails (y compris les performances du modèle) pour chaque version du modèle sur la page de détails du modèle. Au fil du temps, vous pouvez voir comment les performances du modèle évoluent lorsque vous modifiez votre jeu de données d'entraînement.

Vous pouvez créer des versions de modèles à l'aide de la console ou de l'API Amazon Comprehend. Amazon Comprehend propose également de simplifier les tâches associées Volants à la formation et à l'évaluation de nouvelles versions de modèles personnalisés.

Après avoir créé un modèle personnalisé, vous pouvez le partager avec d'autres utilisateurs en autorisant d'autres utilisateurs Comptes AWS à importer une copie de votre modèle.

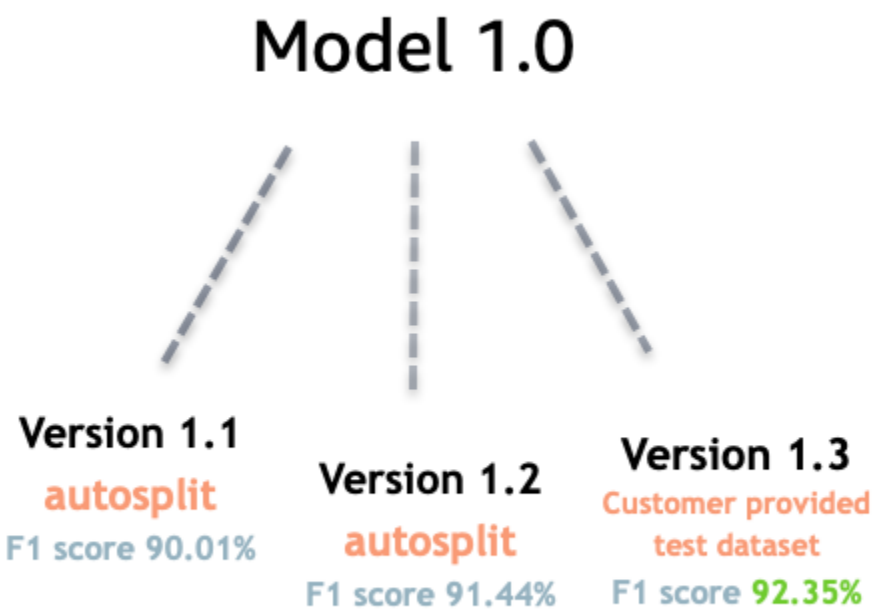
Rubriques

- Versionnage de modèles avec Amazon Comprehend
- Copier des modèles personnalisés entre Comptes AWS

Versionnage de modèles avec Amazon Comprehend

L'intelligence artificielle et l'apprentissage automatique (IA/ML) sont axés sur l'expérimentation rapide. Avec Amazon Comprehend, vous entraînez et créez des modèles que vous utilisez pour mieux comprendre vos données. Grâce au versionnement des modèles, vous pouvez suivre l'historique de votre modélisation et les scores associés aux résultats d'exécution de vos modèles lorsque vous fournissez des ensembles de données plus nombreux ou différents. Vous pouvez utiliser le versionnement avec vos modèles de classification personnalisés ou vos modèles de reconnaissance d'entités personnalisés. En examinant vos différentes versions au fil du temps, vous pouvez avoir un aperçu de leur succès et des paramètres que vous avez utilisés pour atteindre votre niveau de réussite.

Lorsque vous entraînez une nouvelle version d'un modèle de classificateur personnalisé ou d'un modèle de reconnaissance d'entité existant, il vous suffit de créer une nouvelle version à partir de la page des détails du modèle pour que tous les détails soient renseignés pour vous. La nouvelle version portera le même nom que votre modèle précédent (ce que nous appelons le VersionID), mais vous lui donnerez un nom de version unique lors de sa création. Lorsque vous ajoutez de nouvelles versions à un modèle, vous pouvez voir toutes les versions précédentes et leurs détails dans un seul affichage sur la page des détails du modèle. Grâce au versionnement, vous pouvez voir comment les performances du modèle évoluent lorsque vous modifiez votre jeu de données d'entraînement.



Création d'une nouvelle version du classificateur personnalisé (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Personnalisation, puis Classification personnalisée.
3. Dans la liste des classificateurs, choisissez le nom du modèle personnalisé à partir duquel vous souhaitez créer une nouvelle version. La page de détails du modèle personnalisé s'affiche.
4. En haut à droite, sélectionnez Créer un nouveau modèle. Un écran s'ouvre avec des informations préremplies provenant du modèle de classification personnalisé parent.

5. Sous Nom de la version, ajoutez un nom unique à la nouvelle version.
6. Dans les détails de la version, vous pouvez modifier la langue et le nombre d'étiquettes associées à votre nouveau modèle.
7. Dans la section Spécifications des données, configurez la manière dont vous souhaitez fournir les données à votre nouvelle version. Assurez-vous de fournir des données complètes, y compris les documents de votre modèle précédent et vos nouveaux documents. Vous pouvez modifier le mode classificateur (étiquette unique ou étiquette multiple), le format des données (fichier CSV, manifeste augmenté), votre jeu de données d'entraînement et votre jeu de données de test (fractionnement automatique ou configuration de données de test personnalisée).
8. (Facultatif) mettez à jour l'emplacement S3 de vos données de sortie
9. Sous Autorisations d'accès, créez ou utilisez un rôle IAM existant.
10. (Facultatif) Mettez à jour les paramètres de votre VPC
11. (Facultatif) Ajoutez des balises à votre nouvelle version pour suivre les détails.

    Pour plus d'informations sur la création de classificateurs personnalisés, voir Création d'un classificateur personnalisé

Création d'une nouvelle version de reconnaissance d'entités personnalisée (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Personnalisation, puis Reconnaissance d'entité personnalisée.
3. Dans la liste des modèles de reconnaissance, choisissez le nom du système de reconnaissance à partir duquel vous souhaitez créer une nouvelle version. La page de détails s'affiche.
4. En haut à droite, sélectionnez Train new version. Un écran s'ouvre avec des informations préremplies provenant du système de reconnaissance d'entités parent.
5. Sous Nom de la version, ajoutez un nom unique à la nouvelle version.
6. Sous Type d'entité personnalisé, ajoutez les étiquettes personnalisées ou l'étiquette que vous souhaitez que le logiciel de reconnaissance identifie dans votre ensemble de données et sélectionnez Ajouter un type. Choisissez un type d'entité personnalisé dans les annotations ou la liste d'entités que vous avez fournies. Le système de reconnaissance utilisera ensuite tous les types d'entités inclus pour identifier les entités dans l'ensemble de données lors de l'exécution de votre tâche. Chaque type d'entité doit être en majuscules et séparé par un trait de soulignement s'il contient plusieurs mots. Un maximum de 25 types sont autorisés.

7. (Facultatif) Sélectionnez le chiffrement Recognizer pour chiffrer les données du volume de stockage pendant le traitement de votre tâche.
8. Dans la section Données d'entraînement, spécifiez les détails de l'annotation et du format des données (fichier CSV, manifeste augmenté) (étiquette unique ou étiquette multiple), le format des données (CSV, manifeste augmenté), votre jeu de données d'entraînement et votre jeu de données de test (fractionnement automatique ou configuration de données de test personnalisée).
9. (Facultatif) mettez à jour l'emplacement S3 de vos données de sortie
10. Sous Autorisations d'accès, créez ou utilisez un rôle IAM existant.
11. (Facultatif) Mettez à jour les paramètres de votre VPC
12. (Facultatif) Ajoutez des balises à votre nouvelle version pour suivre les détails.

Pour en savoir plus sur les outils de reconnaissance d'entités personnalisés, consultez les sections Reconnaissance d'entités personnalisée et Création d'un outil de reconnaissance d'entités personnalisé à l'aide de la console.

Copier des modèles personnalisés entre Comptes AWS

Les utilisateurs d'Amazon Comprehend peuvent copier des modèles personnalisés entraînés entre eux dans le cadre d'un Comptes AWS processus en deux étapes. Tout d'abord, un utilisateur Compte AWS (compte A) partage un modèle personnalisé qui se trouve dans son compte. Ensuite, un utilisateur d'un autre Compte AWS (compte B) importe le modèle dans son compte. L'utilisateur du compte B n'a pas besoin d'entraîner le modèle et n'a pas besoin de copier (ou d'accéder) aux données d'entraînement ou de test d'origine.

Pour partager un modèle personnalisé dans le compte A, l'utilisateur associe une politique AWS Identity and Access Management (IAM) à une version du modèle. Cette politique autorise une entité du compte B, telle qu'un utilisateur ou un rôle, à importer la version du modèle dans Amazon Comprehend dans son. Compte AWS L'utilisateur du compte B doit importer le modèle dans le même format Région AWS que le modèle d'origine.

Pour importer le modèle dans le compte B, l'utilisateur de ce compte fournit à Amazon Comprehend les informations nécessaires, telles que le nom de ressource Amazon (ARN) du modèle. En important le modèle, cet utilisateur crée un nouveau modèle personnalisé Compte AWS qui reproduit le modèle qu'il a importé. Ce modèle est entièrement formé et prêt pour les tâches d'inférence, telles que la classification de documents ou la reconnaissance d'entités nommées.

La copie d'un modèle personnalisé est utile si :

- Vous appartenez à une organisation qui en utilise plusieurs Comptes AWS. Par exemple, votre organisation peut disposer d'un identifiant Compte AWS pour chaque phase de développement, telle que la construction, le stage, le test et le déploiement. Il peut également être distinct Comptes AWS pour les fonctions commerciales, telles que la science des données et l'ingénierie.
- Votre organisation travaille avec un autre partenaire, tel qu'un AWS partenaire, qui forme des modèles personnalisés dans Amazon Comprehend et vous les fournit en tant que client.

Dans de tels scénarios, vous pouvez rapidement copier un outil de reconnaissance d'entités ou un classificateur de documents personnalisé entraîné de l'un Compte AWS à l'autre. Copier un modèle de cette manière est plus facile que l'alternative, qui consiste à copier des données d'entraînement entre deux Comptes AWS pour entraîner des modèles dupliqués.

Rubriques

- Partage d'un modèle personnalisé avec un autre Compte AWS
- Importation d'un modèle personnalisé depuis un autre Compte AWS

Partage d'un modèle personnalisé avec un autre Compte AWS

Avec Amazon Comprehend, vous pouvez partager vos modèles personnalisés avec d'autres utilisateurs, afin qu'ils puissent les importer dans leurs AWS comptes. Lorsqu'un utilisateur importe l'un de vos modèles personnalisés, il crée un nouveau modèle personnalisé dans son compte. Leur nouveau modèle reproduit celui que vous avez partagé.

Pour partager un modèle personnalisé, vous devez y associer une politique qui autorise d'autres utilisateurs à l'importer. Ensuite, vous fournissez à ces utilisateurs les informations dont ils ont besoin.

> **Note**
>
> Lorsque d'autres utilisateurs importent un modèle personnalisé que vous avez partagé, ils doivent utiliser le même modèle Région AWS (par exemple, USA East (Virginie du Nord)) qui contient votre modèle.

Rubriques

- Avant de commencer

- Politiques basées sur les ressources pour les modèles personnalisés
- Étape 1 : ajouter une politique basée sur les ressources à un modèle personnalisé
- Étape 2 : Fournissez les informations dont les autres ont besoin pour importer

Avant de commencer

Avant de pouvoir partager un modèle, vous devez disposer d'un classificateur personnalisé qualifié ou d'un outil de reconnaissance d'entités personnalisé dans Amazon Comprehend dans votre. Compte AWS Pour plus d'informations sur la formation de modèles personnalisés, consultez Classification personnalisée ouReconnaissance d'entités personnalisée.

Autorisations requises

Déclaration de politique IAM

Avant de pouvoir ajouter une politique basée sur les ressources à un modèle personnalisé, vous devez disposer d'autorisations dans AWS Identity and Access Management (IAM). Une politique doit être attachée à votre utilisateur, groupe ou rôle afin que vous puissiez créer, obtenir et supprimer des modèles de politiques, comme illustré dans l'exemple suivant.

Example Politique IAM pour gérer les politiques basées sur les ressources pour les modèles personnalisés

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:PutResourcePolicy",
    "comprehend:DeleteResourcePolicy",
    "comprehend:DescribeResourcePolicy"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Pour plus d'informations sur la création d'une stratégie IAM, consultez la section Création de politiques IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'attachement d'une politique IAM, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le guide de l'utilisateur IAM.

AWS KMS déclaration de politique clé

Si vous partagez un modèle chiffré, vous devrez peut-être ajouter des autorisations pour AWS KMS. Cette exigence dépend du type de clé KMS que vous utilisez pour chiffrer le modèle dans Amazon Comprehend.

An Clé détenue par AWSest détenu et géré par un AWS service. Si vous utilisez un Clé détenue par AWS, vous n'avez pas besoin d'ajouter d'autorisations pour AWS KMS, et vous pouvez ignorer cette section.

Une clé gérée par le client est une clé que vous créez, détenez et gérez dans votre Compte AWS. Si vous utilisez une clé gérée par le client, vous devez ajouter une déclaration à votre politique en matière de clés KMS.

La déclaration de politique autorise une ou plusieurs entités (telles que des utilisateurs ou des comptes) à effectuer les AWS KMS opérations nécessaires pour déchiffrer le modèle.

Vous utilisez les touches de condition pour éviter le problème de confusion des adjoints. Pour de plus amples informations, veuillez consulter the section called “Prévention du problème de l’adjoint confus entre services”.

Utilisez les clés de condition suivantes dans la politique pour valider les entités qui accèdent à votre clé KMS. Lorsqu'un utilisateur importe le modèle, il AWS KMS vérifie que l'ARN de la version du modèle source correspond à la condition. Si vous n'incluez aucune condition dans la politique, les principaux spécifiés peuvent utiliser votre clé KMS pour déchiffrer n'importe quelle version du modèle :

- aws : SourceArn — Utilisez cette clé de condition avec les `kms:Decrypt` actions `kms:GenerateDataKey` et.
- kms : EncryptionContext — Utilisez cette clé de condition avec les `kms:CreateGrant` actions `kms:GenerateDataKeykms:Decrypt`, et.

Dans l'exemple suivant, la politique autorise l'utilisation Compte AWS `444455556666` de la version 1 du modèle de classificateur spécifié appartenant à. Compte AWS `111122223333`

Example Politique clé KMS pour accéder à une version spécifique d'un modèle de classificateur

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
          "AWS":
                "arn:aws:iam::444455556666:root"
      },
      "Action": [
          "kms:Decrypt",
          "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": {
              "aws:SourceArn":
                "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/classifierName/version/1"
          }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
          "AWS":  "arn:aws:iam::444455556666:root"
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
           "kms:EncryptionContext:aws:comprehend:arn":
              "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/classifierName/version/1"
        }
      }
    }
  ]
}
```

L'exemple de politique suivant autorise l'utilisateur ExampleUser ExampleRoledepuis Compte AWS 444455556666 et vers l'utilisateur Compte AWS 123456789012 à accéder à cette clé KMS via le service Amazon Comprehend.

Example Politique clé de KMS autorisant l'accès au service Amazon Comprehend (alternative 1).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": [
                "arn:aws:iam::444455556666:user/ExampleUser",
                "arn:aws:iam::123456789012:role/ExampleRole"
              ]
        },
        "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": "arn:aws:comprehend:*"

            }
        }
    },
    {
      "Effect": "Allow",
      "Principal": {
         "AWS": [
                "arn:aws:iam::444455556666:user/ExampleUser",
                "arn:aws:iam::123456789012:role/ExampleRole"
              ]
      },
      "Action": "kms:CreateGrant",
      "Resource": "*",
      "Condition": {
          "StringLike": {
              "kms:EncryptionContext:aws:comprehend:arn": "arn:aws:comprehend:*"
          }
      }
    }
  ]
}
```

L'exemple de politique suivant autorise l'accès Compte AWS 444455556666 à cette clé KMS via le service Amazon Comprehend, en utilisant une syntaxe alternative à celle de l'exemple précédent.

Example Politique clé de KMS autorisant l'accès au service Amazon Comprehend (alternative 2).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::444455556666:root"
      },
      "Action": [
          "kms:Decrypt",
          "kms:GenerateDataKey",
          "kms:CreateGrant"
      ],
      "Resource": "*",
      "Condition": {
          "StringLike": {
              "kms:EncryptionContext:aws:comprehend:arn": "arn:aws:comprehend:*"
          }
      }
    }
  ]
}
```

Pour plus d'informations, consultez Politiques de clé dans AWS KMS dans le Guide du développeur AWS Key Management Service .

Politiques basées sur les ressources pour les modèles personnalisés

Avant qu'un utilisateur d'Amazon Comprehend d'un autre utilisateur Compte AWS puisse importer un modèle personnalisé depuis votre AWS compte, vous devez l'autoriser à le faire. Pour les autoriser, vous ajoutez une politique basée sur les ressources à la version du modèle que vous souhaitez partager. Une stratégie basée sur les ressources est une stratégie IAM que vous attachez à une ressource dans. AWS

Lorsque vous associez une politique de ressources à une version de modèle personnalisée, la politique autorise les utilisateurs, les groupes ou les rôles à effectuer l'`comprehend:ImportModel`action sur la version du modèle.

Example Politique basée sur les ressources pour une version de modèle personnalisée

Cet exemple indique les entités autorisées dans l'`Principal`attribut. La ressource « * » fait référence à la version du modèle spécifique à laquelle vous attachez la politique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "comprehend:ImportModel",
      "Resource": "*",
      "Principal": {
        "AWS": [
                "arn:aws:iam::111122223333:root",
                "arn:aws:iam::444455556666:user/ExampleUser",
                "arn:aws:iam::123456789012:role/ExampleRole"
         ]
      }
    }
  ]
}
```

Pour les politiques que vous associez à des modèles personnalisés, `comprehend:ImportModel` c'est la seule action prise en charge par Amazon Comprehend.

Pour plus d'informations sur les politiques basées sur les ressources, consultez les sections Politiques basées sur l'identité et politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Étape 1 : ajouter une politique basée sur les ressources à un modèle personnalisé

Vous pouvez ajouter une politique basée sur les ressources à l'aide de l'API AWS Management Console AWS CLI, ou Amazon Comprehend.

AWS Management Console

Vous pouvez utiliser Amazon Comprehend dans le. AWS Management Console

Pour ajouter une politique basée sur les ressources

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/

2. Dans le menu de navigation de gauche, sous Personnalisation, choisissez la page qui contient votre modèle personnalisé :

   a. Si vous partagez un classificateur de documents personnalisé, choisissez Classification personnalisée.

   b. Si vous partagez un outil de reconnaissance d'entités personnalisé, choisissez Reconnaissance d'entité personnalisée.

3. Dans la liste des modèles, choisissez le nom du modèle pour ouvrir sa page de détails.
4. Sous Versions, choisissez le nom de la version du modèle que vous souhaitez partager.
5. Sur la page des détails de la version, choisissez l'onglet Tags, VPC & Policy.
6. Dans la section Stratégie basée sur les ressources, choisissez Modifier.
7. Sur la page Modifier une politique basée sur les ressources, procédez comme suit :

   a. Dans Nom de la politique, entrez un nom qui vous aidera à reconnaître la politique une fois que vous l'aurez créée.

   b. Sous Autoriser, spécifiez une ou plusieurs des entités suivantes pour les autoriser à importer votre modèle :

Champ	Définition et exemples
Principaux du service	Identifiants principaux de service pour les services qui peuvent accéder à cette version du modèle. Par exemple : comprend.amazonaws.com
Compte AWS IDs	Comptes AWS qui peuvent accéder à cette version du modèle. Autorise tous les utilisateurs appartenant au compte. Par exemple : 111122223333, 123456789012
Entités IAM	ARNs pour les utilisateurs ou les rôles qui peuvent accéder à cette version du modèle. Par exemple : arn:aws:iam:111122223333 : user/ExampleUser, arn:aws:iam::444455556666:role/ExampleRole

8. Sous Partager, vous pouvez copier l'ARN de la version du modèle pour le partager avec la personne qui va importer votre modèle. Lorsque quelqu'un importe un modèle personnalisé à partir d'un modèle différent Compte AWS, l'ARN de la version du modèle est requis.
9. Choisissez Save (Enregistrer). Amazon Comprehend crée votre politique basée sur les ressources et l'associe à votre modèle.

AWS CLI

Pour ajouter une politique basée sur les ressources à un modèle personnalisé avec le AWS CLI, utilisez la PutResourcePolicycommande. La commande d' utilise les paramètres suivants :

- `resource-arn`— L'ARN du modèle personnalisé, y compris la version du modèle.
- `resource-policy`— Un fichier JSON qui définit la politique basée sur les ressources à associer à votre modèle personnalisé.

  Vous pouvez également fournir la politique sous forme de chaîne JSON intégrée. Pour fournir un code JSON valide pour votre politique, placez les noms et les valeurs d'attributs entre guillemets. Si le corps du JSON est également placé entre guillemets, vous évitez les guillemets figurant dans la politique.
- `policy-revision-id`— L'ID de révision attribué par Amazon Comprehend à la politique que vous mettez à jour. Si vous créez une nouvelle politique sans version précédente, n'utilisez pas ce paramètre. Amazon Comprehend crée l'ID de révision pour vous.

Example Ajoutez une politique basée sur les ressources à un modèle personnalisé à l'aide de la commande **`put-resource-policy`**

Cet exemple définit une politique dans un fichier JSON nommé PolicyFile.json et associe la politique à un modèle. Le modèle est la version v2 d'un classificateur nommé mycf1.

```
$ aws comprehend put-resource-policy \
> --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/mycf1/
version/v2 \
> --resource-policy file://policyFile.json \
> --policy-revision-id revision-id
```

Le fichier JSON de la politique de ressources contient le contenu suivant :

- Action — La politique autorise les principaux nommés à utiliser. `comprehend:ImportModel`
- Ressource — L'ARN du modèle personnalisé. La ressource « * » fait référence à la version du modèle que vous spécifiez dans la `put-resource-policy` commande.
- Principal — La politique autorise l'utilisateur à partir du Compte AWS 444455556666 et tous les utilisateurs `jane` à partir du 123456789012. Compte AWS

```
{
"Version":"2012-10-17",
 "Statement":[
    {"Sid":"ResourcePolicyForImportModel",
     "Effect":"Allow",
     "Action":["comprehend:ImportModel"],
     "Resource":"*",
     "Principal":
         {"AWS":
            ["arn:aws:iam::444455556666:user/jane",
             "123456789012"]
         }
   }
 ]
}
```

API Amazon Comprehend

Pour ajouter une politique basée sur les ressources à un modèle personnalisé à l'aide de l'API Amazon Comprehend, utilisez PutResourcePolicyl'opération API.

Vous pouvez également ajouter une politique à un modèle personnalisé dans la demande d'API qui crée le modèle. Pour ce faire, fournissez le JSON de politique pour le ModelPolicy paramètre lorsque vous soumettez une CreateEntityRecognizerdemande CreateDocumentClassifieror.

Étape 2 : Fournissez les informations dont les autres ont besoin pour importer

Maintenant que vous avez ajouté la politique basée sur les ressources à votre modèle personnalisé, vous avez autorisé les autres utilisateurs d'Amazon Comprehend à importer votre modèle dans leur. Comptes AWS Toutefois, avant de pouvoir les importer, vous devez leur fournir les informations suivantes :

- Le nom de ressource Amazon (ARN) de la version du modèle.

- Celui Région AWS qui contient le modèle. Toute personne qui importe votre modèle doit l'utiliser Région AWS .
- Si le modèle est crypté, et si c'est le cas, quel est le type de AWS KMS clé que vous utilisez : Clé détenue par AWS ou clé gérée par le client.
- Si votre modèle est chiffré à l'aide d'une clé gérée par le client, vous devez fournir l'ARN de la clé KMS. Toute personne qui importe votre modèle doit inclure l'ARN d'un rôle de service IAM dans son Compte AWS. Ce rôle autorise Amazon Comprehend à utiliser la clé KMS pour déchiffrer le modèle lors de l'importation.

Pour plus d'informations sur la façon dont les autres utilisateurs importent votre modèle, consultezImportation d'un modèle personnalisé depuis un autre Compte AWS.

Importation d'un modèle personnalisé depuis un autre Compte AWS

Dans Amazon Comprehend, vous pouvez importer un modèle personnalisé qui se trouve dans un autre. Compte AWS Lorsque vous importez un modèle, vous créez un nouveau modèle personnalisé dans votre compte. Votre nouveau modèle personnalisé est une copie entièrement entraînée du modèle que vous avez importé.

Rubriques

- Avant de commencer
- Importation d'un modèle personnalisé

Avant de commencer

Avant de pouvoir importer un modèle personnalisé depuis un autre Compte AWS, assurez-vous que la personne qui a partagé le modèle avec vous effectue les opérations suivantes :

- Vous autorise à effectuer l'importation. Cette autorisation est accordée dans la politique basée sur les ressources attachée à la version du modèle. Pour de plus amples informations, veuillez consulter Politiques basées sur les ressources pour les modèles personnalisés.
- Vous fournit les informations suivantes :
  - Le nom de ressource Amazon (ARN) de la version du modèle.
  - Celui Région AWS qui contient le modèle. Vous devez utiliser le même Région AWS lors de l'importation.
  - Si le modèle est chiffré à l'aide d'une AWS KMS clé et, dans l'affirmative, le type de clé utilisé.

Si le modèle est chiffré, vous devrez peut-être prendre des mesures supplémentaires, en fonction du type de clé KMS utilisé :

- Clé détenue par AWS— Ce type de clé KMS est détenu et géré par AWS. Si le modèle est chiffré avec un Clé détenue par AWS, aucune étape supplémentaire n'est nécessaire.
- Clé gérée par le client — Ce type de clé KMS est créé, détenu et géré par un AWS client dans son Compte AWS. Si le modèle est chiffré à l'aide d'une clé gérée par le client, la personne qui a partagé le modèle doit :
  - Vous autoriser à déchiffrer le modèle. Cette autorisation est accordée dans la politique de clé KMS pour la clé gérée par le client. Pour de plus amples informations, veuillez consulter AWS KMS déclaration de politique clé.
  - Indiquez l'ARN de la clé gérée par le client. Vous utilisez cet ARN lorsque vous créez un rôle de service IAM. Ce rôle autorise Amazon Comprehend à utiliser la clé KMS pour déchiffrer le modèle.

Autorisations requises

Avant de pouvoir importer un modèle personnalisé, vous ou votre administrateur devez autoriser les actions requises dans AWS Identity and Access Management (IAM). En tant qu'utilisateur d'Amazon Comprehend, vous devez être autorisé à importer par une déclaration de politique IAM. Si le chiffrement ou le déchiffrement est requis lors de l'importation, Amazon Comprehend doit être autorisé à utiliser les AWS KMS clés nécessaires.

Déclaration de politique IAM

Votre utilisateur, groupe ou rôle doit être associé à une politique autorisant l'`ImportModel`action, comme illustré dans l'exemple suivant.

Example Politique IAM pour importer un modèle personnalisé

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:ImportModel"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Pour plus d'informations sur la création d'une stratégie IAM, consultez la section Création de politiques IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur l'attachement d'une politique IAM, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le guide de l'utilisateur IAM.

Rôle de service IAM pour le chiffrement AWS KMS

Lorsque vous importez un modèle personnalisé, vous devez autoriser Amazon Comprehend à utiliser des AWS KMS clés dans l'un des cas suivants :

- Vous importez un modèle personnalisé chiffré à l'aide d'une clé gérée par le client AWS KMS. Dans ce cas, Amazon Comprehend doit accéder à la clé KMS afin de pouvoir déchiffrer le modèle lors de l'importation.
- Vous souhaitez chiffrer le nouveau modèle personnalisé que vous créez lors de l'importation, et vous souhaitez utiliser une clé gérée par le client. Dans ce cas, Amazon Comprehend a besoin d'accéder à votre clé KMS pour pouvoir chiffrer le nouveau modèle.

Pour autoriser Amazon Comprehend à utiliser ces AWS KMS clés, vous devez créer un rôle de service IAM. Ce type de rôle IAM permet à un AWS service d'accéder aux ressources d'autres services en votre nom. Pour plus d'informations sur les rôles de service, consultez la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le Guide de l'utilisateur IAM.

Si vous utilisez la console Amazon Comprehend pour importer, Amazon Comprehend peut créer le rôle de service pour vous. Dans le cas contraire, vous devez créer un rôle de service dans IAM avant de procéder à l'importation.

Le rôle de service IAM doit disposer d'une politique d'autorisations et d'une politique de confiance, comme le montrent les exemples suivants.

Example stratégie d'autorisation

La politique d'autorisation suivante autorise les AWS KMS opérations utilisées par Amazon Comprehend pour chiffrer et déchiffrer des modèles personnalisés. Il donne accès à deux clés KMS :

- Une clé KMS Compte AWS contient le modèle à importer. Il a été utilisé pour chiffrer le modèle, et Amazon Comprehend l'utilise pour le déchiffrer lors de l'importation.
- L'autre clé KMS se trouve dans le Compte AWS qui importe le modèle. Amazon Comprehend utilise cette clé pour chiffrer le nouveau modèle personnalisé créé par l'importation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "kms:CreateGrant"
        ],
        "Resource": [
            "arn:aws:kms:us-west-2:111122223333:key/key-id",
            "arn:aws:kms:us-west-2:444455556666:key/key-id"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt",
            "kms:GenerateDatakey"
        ],
        "Resource": [
            "arn:aws:kms:us-west-2:111122223333:key/key-id",
            "arn:aws:kms:us-west-2:444455556666:key/key-id"
        ],
        "Condition": {
            "StringEquals": {
                "kms:ViaService": [
                    "s3.us-west-2.amazonaws.com"
                ]
            }
        }
    }
  ]
}
```

Example politique d'approbation

La politique de confiance suivante permet à Amazon Comprehend d'assumer le rôle et d'obtenir ses autorisations. Il permet au directeur du `comprehend.amazonaws.com` service d'effectuer l'`sts:AssumeRole`opération. Pour éviter toute confusion chez les adjoints, vous limitez la portée de l'autorisation en utilisant une ou plusieurs clés contextuelles de conditions globales. Pour`aws:SourceAccount`, spécifiez l'ID de compte de l'utilisateur qui importe le modèle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "444455556666"
        }
      }
    }
  ]
}
```

Importation d'un modèle personnalisé

Vous pouvez importer un modèle personnalisé à l'aide de l' AWS Management Console API AWS CLI, ou Amazon Comprehend.

AWS Management Console

Vous pouvez utiliser Amazon Comprehend dans le. AWS Management Console

Pour importer un modèle personnalisé

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de navigation de gauche, sous Personnalisation, choisissez la page correspondant au type de modèle que vous importez :

   a. Si vous importez un classificateur de documents personnalisé, choisissez Classification personnalisée.

   b. Si vous importez un outil de reconnaissance d'entité personnalisé, choisissez Reconnaissance d'entité personnalisée.
3. Choisissez la version d'importation.
4. Sur la page Importer la version du modèle, entrez les informations suivantes :

- ARN de la version du modèle : ARN de la version du modèle à importer.
- Nom du modèle : nom personnalisé du nouveau modèle créé par l'importation.
- Nom de version : nom personnalisé pour la nouvelle version du modèle créée par l'importation.

5. Pour le chiffrement du modèle, choisissez le type de clé KMS à utiliser pour chiffrer le nouveau modèle personnalisé que vous créez lors de l'importation :

   - Utiliser AWS une clé propre : Amazon Comprehend chiffre votre modèle à l'aide d'une clé in AWS Key Management Service (AWS KMS) créée, gérée et utilisée en votre nom par. AWS
   - Choisissez une autre AWS KMS clé (avancée) — Amazon Comprehend chiffre votre modèle à l'aide d'une clé gérée par le client dans laquelle vous gérez. AWS KMS

     Si vous choisissez cette option, sélectionnez une clé KMS qui se trouve dans votre Compte AWS, ou créez-en une nouvelle en choisissant Créer une AWS KMS clé.

6. Dans la section Accès au service, accordez à Amazon Comprehend l'accès à toutes AWS KMS les clés dont il a besoin pour :

   - Déchiffrez le modèle personnalisé que vous importez.
   - Chiffrez le nouveau modèle personnalisé que vous créez lors de l'importation.

   Vous accordez l'accès avec un rôle de service IAM qui permet à Amazon Comprehend d'utiliser les clés KMS.

   Pour le rôle de service, effectuez l'une des opérations suivantes :

   - Si vous souhaitez utiliser un rôle de service existant, choisissez Utiliser un rôle IAM existant. Sélectionnez-le ensuite sous Nom du rôle.
   - Si vous souhaitez qu'Amazon Comprehend crée un rôle pour vous, choisissez Create an IAM role.

7. Si vous avez choisi de demander à Amazon Comprehend de créer le rôle pour vous, procédez comme suit :

   a. Dans Nom du rôle, entrez un suffixe de nom de rôle qui vous aidera à reconnaître le rôle ultérieurement.

   b. Pour l'ARN de la clé KMS source, entrez l'ARN de la clé KMS utilisée pour chiffrer le modèle que vous importez. Amazon Comprehend utilise cette clé pour déchiffrer le modèle lors de l'importation.

8. (Facultatif) Dans la section Balises, vous pouvez ajouter des balises au nouveau modèle personnalisé que vous créez par importation. Pour plus d'informations sur le balisage de modèles personnalisés, consultezMarquer une nouvelle ressource.
9. Choisissez Confirmer.

AWS CLI

Vous pouvez utiliser Amazon Comprehend en exécutant des commandes à l'aide du. AWS CLI

Example Commande d'importation du modèle

Pour importer un modèle personnalisé, utilisez la `import-model` commande :

```
$ aws comprehend import-model \
> --source-model arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/bar \
> --model-name importedDocumentClassifier \
> --version-name versionOne \
> --data-access-role-arn arn:aws:iam::444455556666:role/comprehendAccessRole \
> --model-kms-key-id kms-key-id
```

Cet exemple utilise les paramètres suivants :

- `source-model`— L'ARN du modèle personnalisé à importer.
- `model-name`— Un nom personnalisé pour le nouveau modèle créé par l'importation.
- `version-name`— Un nom personnalisé pour la nouvelle version du modèle créée par l'importation.
- `data-access-role-arn`— L'ARN du rôle de service IAM qui permet à Amazon Comprehend d'utiliser les clés AWS KMS nécessaires pour chiffrer ou déchiffrer le modèle personnalisé.
- `model-kms-key-id`— L'ARN ou l'ID de la clé KMS qu'Amazon Comprehend utilise pour chiffrer le modèle personnalisé que vous créez avec cette importation. Cette clé doit se trouver AWS KMS dans votre Compte AWS.

API Amazon Comprehend

Pour importer un modèle personnalisé à l'aide de l'API Amazon Comprehend, utilisez l'action ImportModelAPI.

Volants

Un volant Amazon Comprehend simplifie le processus d'amélioration d'un modèle personnalisé au fil du temps. Vous pouvez utiliser un volant pour orchestrer les tâches associées à la formation et à l'évaluation de nouvelles versions de modèles personnalisés. Les volants prennent en charge les modèles personnalisés en texte brut pour une classification personnalisée et une reconnaissance d'entités personnalisée.

Rubriques

- Vue d'ensemble du volant
- Lacs de données Flywheel
- Autorisations et politiques IAM
- Configuration des volants à l'aide de la console
- Configuration des volants à l'aide de l'API
- Configuration des ensembles de données
- Itérations du volant
- Utilisation de volants pour l'analyse

Vue d'ensemble du volant

Un volant est une ressource Amazon Comprehend qui orchestre la formation et l'évaluation des nouvelles versions d'un modèle personnalisé. Vous pouvez créer un volant pour utiliser un modèle entraîné existant, ou Amazon Comprehend peut créer et entraîner un nouveau modèle pour le volant. Utilisez des volants avec des modèles personnalisés en texte brut pour une classification personnalisée ou une reconnaissance d'entités personnalisée.

Vous pouvez configurer et gérer les volants à l'aide de la console ou de l'API Amazon Comprehend. Vous pouvez également configurer les volants à l'aide AWS CloudFormation de.

Lorsque vous créez un volant, Amazon Comprehend crée un lac de données dans votre compte. Le lac de données stocke et gère toutes les données relatives au volant, telles que les données d'entraînement et les données de test pour toutes les versions du modèle.

Vous définissez la version du modèle actif comme étant la version du modèle de volant que vous souhaitez utiliser pour les tâches d'inférence ou les points de terminaison Amazon Comprehend.

Au départ, le volant contient une version du modèle. Au fil du temps, lorsque vous entraînez de nouvelles versions de modèles, vous sélectionnez la version la plus performante comme version active. Lorsqu'un utilisateur spécifie l'ARN du volant pour exécuter une tâche d'inférence, Amazon Comprehend exécute la tâche en utilisant la version du modèle actif du volant.

Régulièrement, vous obtenez de nouvelles données étiquetées (données d'entraînement ou données de test) pour le modèle. Vous mettez de nouvelles données à la disposition du volant en créant un ou plusieurs ensembles de données. Un jeu de données contient des données d'entrée pour l'entraînement ou le test du modèle personnalisé associé à un volant. Amazon Comprehend télécharge les données d'entrée dans le lac de données du volant.

Pour intégrer les nouveaux ensembles de données dans votre modèle personnalisé, vous devez créer et exécuter une itération en volant. Une itération en volant est un flux de travail qui utilise les nouveaux ensembles de données pour évaluer la version active du modèle et pour entraîner une nouvelle version du modèle. Sur la base des statistiques relatives aux versions existantes et nouvelles du modèle, vous pouvez décider de promouvoir ou non la nouvelle version du modèle en tant que version active.

Vous pouvez utiliser la version du modèle actif au volant pour exécuter des analyses personnalisées (tâches en temps réel ou asynchrones). Pour utiliser le modèle de volant à des fins d'analyse en temps réel, vous devez créer un point de terminaison pour le volant.

Il n'y a pas de frais supplémentaires pour l'utilisation des volants. Toutefois, lorsque vous exécutez une itération en volant, vous devez payer les frais standard pour l'entraînement d'une nouvelle version du modèle et le stockage des données du modèle. Pour obtenir des informations détaillées sur les prix, consultez Amazon Comprehend Pricing.

Rubriques

- Ensembles de données Flywheel
- Création d'un volant
- États du volant
- Itérations du volant

Ensembles de données Flywheel

Pour ajouter de nouvelles données étiquetées à un volant, vous devez créer un jeu de données. Vous configurez chaque ensemble de données en tant que données d'entraînement ou données de test. Vous associez le jeu de données à un volant spécifique et à un modèle personnalisé.

Une fois que vous avez créé un ensemble de données, Amazon Comprehend télécharge les données dans le lac de données du volant. Pour de plus amples informations, veuillez consulter Lacs de données Flywheel.

Création d'un volant

Lorsque vous créez un volant, vous pouvez associer le volant à un modèle entraîné existant, ou le volant peut créer un nouveau modèle.

Lorsque vous créez un volant à partir d'un modèle existant, vous spécifiez la version du modèle actif. Amazon Comprehend copie les données d'entraînement et les données de test du modèle dans le data lake du volant. Assurez-vous que les données d'entraînement et de test du modèle se trouvent au même emplacement Amazon S3 que lorsque vous avez créé le modèle.

Pour créer un volant pour un nouveau modèle, vous devez fournir un jeu de données pour les données d'entraînement (et un ensemble de données facultatif pour les données de test) lorsque vous créez le volant. Lorsque vous lancez le volant pour créer la première itération du volant, le volant entraîne le nouveau modèle.

Lorsque vous entraînez un modèle personnalisé, vous spécifiez une liste d'étiquettes personnalisées (classification personnalisée) ou d'entités personnalisées (reconnaissance d'entités personnalisée) que le modèle doit reconnaître. Notez les points importants suivants concernant les étiquettes/entités personnalisées :

- Lorsque vous créez un volant pour un nouveau modèle, la liste des étiquettes/entités que vous fournissez lors de la création du volant est la liste finale du volant.
- Lorsque vous créez un volant à partir d'un modèle existant, la liste des étiquettes/entités associées à ce modèle devient la liste finale du volant.
- Si vous associez un nouveau jeu de données au volant et que cet ensemble de données en contient d'autreslabels/entities, Amazon Comprehend ignores the new labels/entities.
- Vous pouvez consulter la liste des étiquettes/entités d'un volant à l'aide de l'DescribeFlywheelopération API.

> **Note**
>
> Pour une classification personnalisée, Amazon Comprehend renseigne la liste des étiquettes une fois que le statut du volant devient ACTIF. Attendez que le volant soit actif avant d'appeler l'opération DescribeFlywheel API.

États du volant

Un volant passe d'un état à l'autre :

- CRÉATION - Amazon Comprehend est en train de créer les ressources du volant. Vous pouvez effectuer des opérations de lecture sur le volant, telles que`DescribeFlywheel`.
- ACTIF - Le volant est actif. Vous pouvez déterminer si une itération au volant est en cours et consulter le statut de l'itération. Vous pouvez effectuer des actions de lecture sur le volant et des actions telles que `DeleteFlywheel` et`UpdateFlywheel`.
- MISE À JOUR - Amazon Comprehend est en train de mettre à jour le volant. Vous pouvez effectuer des opérations de lecture sur le volant.
- SUPPRESSION - Amazon Comprehend est en train de supprimer le volant. Vous pouvez effectuer des opérations de lecture sur le volant.
- ÉCHEC : l'opération de création du volant a échoué.

Une fois qu'Amazon Comprehend a supprimé un volant, vous conservez l'accès à toutes les données du modèle dans le lac de données du volant. Amazon Comprehend supprime toutes les métadonnées internes requises pour gérer les ressources du volant. Amazon Comprehend supprime également les ensembles de données associés à ce volant (les données du modèle sont enregistrées dans le lac de données).

Itérations du volant

Lorsque vous obtenez de nouvelles données d'entraînement ou de test pour un modèle de volant, vous créez un ou plusieurs nouveaux ensembles de données pour télécharger les nouvelles données dans le lac de données du volant.

Vous lancez ensuite le volant pour créer une nouvelle itération du volant. L'itération du volant évalue la version active actuelle du modèle à l'aide des nouvelles données et stocke les résultats dans le lac de données. Le volant crée et entraîne également une nouvelle version du modèle.

Si le nouveau modèle présente de meilleures performances que la version du modèle actif actuel, vous pouvez promouvoir la nouvelle version du modèle en tant que version du modèle actif. Vous pouvez utiliser la console ou l'opération UpdateFlywheelAPI pour mettre à jour la version du modèle actif.

Lacs de données Flywheel

Lorsque vous créez un volant, Amazon Comprehend crée un lac de données dans votre compte contenant toutes les données relatives au volant, telles que les données d'entrée et de sortie requises pour les versions du modèle.

Amazon Comprehend crée le lac de données à l'emplacement Amazon S3 que vous spécifiez lorsque vous créez le volant. Vous pouvez spécifier l'emplacement en tant que compartiment Amazon S3 ou en tant que nouveau dossier dans un compartiment Amazon S3.

Structure des dossiers du lac de données

Lorsqu'Amazon Comprehend crée le lac de données, il définit la structure de dossiers suivante dans l'emplacement Amazon S3.

> **Warning**
>
> Amazon Comprehend gère l'organisation et le contenu des dossiers du lac de données. Utilisez toujours les opérations de l'API Amazon Comprehend pour modifier les dossiers des lacs de données, sinon votre volant risque de ne pas fonctionner correctement.

```
Document Pool
Annotations Pool
Staging
Model Datasets
  (data for each version of the model)
  VersionID-1
    Training
    Test
    ModelStats
  VersionID-2
    Training
    Test
    ModelStats
```

Pour consulter l'évaluation de la formation d'une version du modèle, effectuez les opérations suivantes :

1. Ouvrez le dossier intitulé Model Datasets au niveau racine du lac de données. Ce dossier contient un sous-dossier pour chaque version du modèle.
2. Ouvrez le dossier correspondant à la version du modèle qui vous intéresse.
3. Ouvrez le dossier nommé ModelStatspour afficher les statistiques du modèle.

Gestion des lacs de données

Amazon Comprehend exécute les tâches suivantes pour gérer le lac de données en votre nom :

- Définit la structure des dossiers du lac de données et intègre les ensembles de données dans les dossiers appropriés.
- Gère les documents d'entrée (tels que les fichiers texte et les fichiers d'annotation) nécessaires à l'entraînement du modèle.
- Gère les données de sortie d'entraînement et d'évaluation associées à chaque version du modèle.
- Gère le chiffrement des fichiers stockés dans le lac de données.

Amazon Comprehend effectue toutes les opérations de création et de mise à jour des données pour le lac de données. Vous conservez un accès complet aux données du lac de données. Par exemple :

- Vous avez un accès complet au contenu du lac de données.
- Le lac de données reste disponible une fois que vous avez supprimé le volant.
- Vous pouvez configurer les journaux d'accès pour le compartiment Amazon S3 qui contient le lac de données.
- Vous pouvez fournir des clés de chiffrement pour les données. Vous les spécifiez lorsque vous créez le volant.

Nous recommandons les bonnes pratiques suivantes :

- N'ajoutez pas manuellement vos propres dossiers ou fichiers dans le lac de données. Ne modifiez ni ne supprimez aucun fichier dans le lac de données.
- Utilisez toujours les opérations de création et de mise à jour d'Amazon Comprehend pour ajouter ou modifier des données dans le lac de données. Par exemple, `CreateDataset` à utiliser pour fournir des données de formation ou de test et `StartFlywheelIteration` pour générer des données d'évaluation pour les versions du modèle.

- La structure du lac de données peut évoluer au fil du temps. Ne créez pas de scripts ou de programmes en aval qui s'appuient explicitement sur la structure du lac de données.
- Lorsque vous indiquez l'emplacement du lac de données pour le volant, nous vous recommandons de créer un préfixe commun pour les données relatives à tous les volants ou d'utiliser un préfixe différent pour chaque volant. Nous ne recommandons pas d'utiliser la trajectoire complète du lac de données d'un volant comme préfixe pour un autre volant.

Autorisations et politiques IAM

Vous configurez les politiques et autorisations suivantes pour utiliser les volants :

- the section called “Configuration des autorisations utilisateur IAM”pour que les utilisateurs puissent accéder aux opérations du volant.
- (Facultatif) the section called “Configurer les autorisations pour les AWS KMS clés” pour le lac de données.
- the section called “Création d'un rôle d'accès aux données”qui autorise Amazon Comprehend à accéder au lac de données.

Configuration des autorisations utilisateur IAM

Pour utiliser les fonctionnalités du volant, ajoutez des politiques d'autorisation appropriées à vos identités AWS Identity and Access Management (IAM) (utilisateurs, groupes et rôles).

L'exemple suivant montre la politique d'autorisation permettant de créer des ensembles de données, de créer et de gérer des volants et de faire fonctionner le volant.

Example Politique IAM pour gérer les volants

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:CreateFlywheel",
    "comprehend:DeleteFlywheel",
    "comprehend:UpdateFlywheel",
    "comprehend:ListFlywheels",
    "comprehend:DescribeFlywheel",
    "comprehend:CreateDataset",
    "comprehend:DescribeDataset",
```

```
    "comprehend:ListDatasets",
    "comprehend:StartFlywheelIteration",
    "comprehend:DescribeFlywheelIteration",
    "comprehend:ListFlywheelIterationHistory"
  ],
  "Resource": "*"
}
```

Pour plus d'informations sur la création de politiques IAM pour Amazon Comprehend, consultez. Comment Amazon Comprehend fonctionne avec IAM

Configurer les autorisations pour les AWS KMS clés

Si vous utilisez des AWS KMS clés pour vos données dans le lac de données, configurez les autorisations requises. Pour plus d'informations, voirAutorisations requises pour utiliser le chiffrement KMS.

Création d'un rôle d'accès aux données

Vous créez un rôle d'accès aux données dans IAM pour Amazon Comprehend afin d'accéder aux données du volant dans le lac de données. Si vous utilisez la console pour créer un volant, le système peut éventuellement créer un nouveau rôle à cette fin. Pour de plus amples informations, veuillez consulter Autorisations basées sur les rôles requises pour les opérations asynchrones.

Configuration des volants à l'aide de la console

Vous pouvez utiliser la console Amazon Comprehend pour créer, mettre à jour et supprimer des volants.

Lorsque vous créez un volant, Amazon Comprehend crée un lac de données contenant toutes les données dont le volant a besoin, telles que les données d'entraînement et les données de test pour chaque version du modèle.

Lorsque vous supprimez un volant, Amazon Comprehend ne supprime pas le lac de données ni le modèle associé au volant.

Consultez les informations de la section Création d'un volant avant de créer un nouveau volant.

Rubriques

- Créez un volant
- Mettre à jour un volant
- Supprimer un volant

Créez un volant

Lorsque vous créez un volant, les champs de configuration obligatoires varient selon que le volant est destiné à un modèle personnalisé existant ou à un nouveau modèle.

Pour créer un volant

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Flywheels.
3. Dans le tableau Flywheels, choisissez Create new flywheel.
4. Sous Nom du volant, entrez le nom du volant.
5. (Facultatif) Pour créer un volant pour un modèle existant, configurez les champs sous Version du modèle actif.
    a. Dans la liste déroulante Modèle, sélectionnez un modèle
    b. Dans la liste déroulante Version, sélectionnez la version du modèle.
6. (Facultatif) Pour créer un nouveau modèle de classificateur pour le volant, sous Type de modèle personnalisé, choisissez une classification personnalisée et configurez les paramètres dans les étapes suivantes.
    a. Sous Langue, sélectionnez la langue du modèle.
    b. En mode classificateur, choisissez le mode à étiquette unique ou le mode à étiquettes multiples.
    c. Sous Étiquettes personnalisées, entrez une ou plusieurs étiquettes personnalisées à utiliser pour entraîner le modèle. Chaque étiquette doit correspondre à l'une des classes figurant dans les données d'entraînement que vous avez saisies.
7. (Facultatif) Pour créer un nouveau modèle de reconnaissance d'entité pour le volant, sous Type de modèle personnalisé, choisissez un modèle de reconnaissance d'entité personnalisé et configurez les paramètres dans les étapes suivantes.
    a. Sous Langue, sélectionnez la langue du modèle.

b. Sous Type d'entité personnalisé, entrez jusqu'à 25 entités personnalisées à utiliser pour entraîner le modèle. Chaque étiquette doit correspondre à l'un des types d'entités figurant dans vos données d'entraînement d'entrée.

   Pour créer plusieurs étiquettes, effectuez les étapes suivantes à plusieurs reprises.

   i. Entrez une étiquette personnalisée. L'étiquette doit être entièrement en majuscules. Utilisez un trait de soulignement pour séparer les mots de l'étiquette.

   ii. Choisissez Ajouter un type.

   Pour supprimer l'une des étiquettes que vous avez ajoutées, choisissez X à droite du nom de l'étiquette.

8. Configurez vos choix pour le chiffrement des volumes, le chiffrement des modèles et le chiffrement des lacs de données. Pour chacune d'entre elles, choisissez d'utiliser une clé KMS AWS détenue ou une clé que vous êtes autorisé à utiliser.

   - Si vous utilisez une AWS clé KMS que vous possédez, il n'y a aucun paramètre supplémentaire.
   - Si vous utilisez une autre clé existante, dans le champ ARN de la clé KMS, entrez l'ARN de l'ID de clé.
   - Si vous souhaitez créer une nouvelle clé, choisissez Create an AWS KMS key.

   Pour plus d'informations sur la création et l'utilisation de clés KMS et le chiffrement associé, consultez AWS Key Management Service.

   a. Configurez la clé de chiffrement du volume. Amazon Comprehend utilise cette clé pour chiffrer les données du volume de stockage pendant le traitement de votre tâche. Choisissez d'utiliser une clé KMS que vous AWS possédez ou une clé que vous êtes autorisé à utiliser.

   b. Configurez la clé de chiffrement du modèle. Amazon Comprehend utilise cette clé pour chiffrer les données du modèle pour cette version du modèle.

9. Configurez l'emplacement du lac de données. Pour de plus amples informations, veuillez consulter Gestion des lacs de données.

10. (Facultatif) Configurez la clé de chiffrement Data Lake. Amazon Comprehend utilise cette clé pour chiffrer tous les fichiers du lac de données.

11. (Facultatif) Configurez les paramètres VPC. Entrez l'ID VPC sous VPC ou choisissez-le dans la liste déroulante.

    1. Choisissez le sous-réseau sous Sous-réseaux (s). Après avoir sélectionné le premier sous-réseau, vous pouvez en choisir d'autres.
    2. Sous Groupe (s) de sécurité, choisissez le groupe de sécurité à utiliser si vous en avez spécifié un. Après avoir sélectionné le premier groupe de sécurité, vous pouvez en choisir d'autres.

12. Configurez les autorisations d'accès au service.

    1. Si vous sélectionnez Utiliser un rôle IAM existant, sélectionnez le nom du rôle dans la liste déroulante.
    2. Si vous sélectionnez Create an IAM role, Amazon Comprehend crée un nouveau rôle. La console affiche les autorisations configurées par Amazon Comprehend pour le rôle. Sous Nom du rôle, entrez un nom descriptif pour le rôle.

13. (Facultatif) Configurez les paramètres des balises. Pour ajouter une balise, entrez une paire clé-valeur sous Balises. Choisissez Ajouter une balise. Pour supprimer cette paire avant de créer le volant, choisissez Supprimer le tag. Pour de plus amples informations, veuillez consulter Balisage de vos ressources .

14. Sélectionnez Créer.

Mettre à jour un volant

Vous pouvez configurer le nom du volant, l'emplacement du lac de données, le type de modèle et la configuration du modèle uniquement lorsque vous créez le volant.

Lorsque vous mettez à jour un volant, vous pouvez spécifier un modèle différent si le type de modèle et les options de configuration sont les mêmes que ceux du modèle actuel. Vous pouvez configurer une nouvelle version du modèle actif. Vous pouvez également mettre à jour les détails du chiffrement, les autorisations d'accès aux services et les paramètres VPC.

Pour mettre à jour un volant

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Flywheels.
3. Dans le tableau Flywheels, choisissez le volant à mettre à jour.

4. Sous Version du modèle actif, choisissez un modèle dans la liste déroulante Modèle et choisissez une version du modèle.

   Le formulaire renseigne le type de modèle et la configuration du modèle.
5. (Facultatif) Configurez les paramètres de chiffrement du volume et du modèle.
6. (Facultatif) Configurez les paramètres de chiffrement du Data Lake.
7. Configurez les autorisations d'accès au service.
8. (Facultatif) Configurez les paramètres VPC.
9. (Facultatif) Configurez les paramètres des balises.
10. Choisissez Save (Enregistrer).

Supprimer un volant

Pour supprimer un volant

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Flywheels.
3. Dans le tableau Volants, choisissez le volant à supprimer.
4. Sélectionnez Supprimer.

Configuration des volants à l'aide de l'API

Vous pouvez utiliser l'API Amazon Comprehend pour créer, mettre à jour et supprimer des volants.

Lorsque vous créez un volant, Amazon Comprehend crée un lac de données contenant toutes les données dont le volant a besoin, telles que les données d'entraînement et les données de test pour chaque version du modèle.

Lorsque vous supprimez un volant, Amazon Comprehend ne supprime pas le lac de données ni le modèle associé au volant.

L'opération de suppression du volant échoue si le volant exécute une itération ou crée un jeu de données.

Consultez les informations de la section Création d'un volant avant de créer un nouveau volant.

Création d'un volant pour un modèle existant

Utilisez cette CreateFlywheelopération pour créer un volant d'inertie pour un modèle existant.

Example

```
aws comprehend create-flywheel  \
    --flywheel-name "myFlywheel2"  \
    --active-model-arn  "modelArn"  \
    --data-access-role-arn   arn:aws::iam::111122223333:role/testFlywheelDataAccess \
    --data-lake-s3-uri": "https://s3-bucket-endpoint"   \
```

Si l'opération est réussie, la réponse inclut l'ARN du volant.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Création d'un volant pour un nouveau modèle

Utilisez cette CreateFlywheelopération pour créer un volant d'inertie pour un nouveau modèle de classification personnalisé.

Example

```
aws comprehend create-flywheel \
    --flywheel-name "myFlywheel2" \
    --data-access-role-arn  arn:aws::iam::111122223333:role/testFlywheelDataAccess \
    --model-type "DOCUMENT_CLASSIFIER" \
    --data-lake-s3-uri  "s3Uri"  \
    --task-config  file://taskConfig.json
```

Le fichier TaskConfig.json contient le contenu suivant.

```
{
    "LanguageCode": "en",
    "DocumentClassificationConfig": {
        "Mode": "MULTI_LABEL",
```

```
        "Labels": ["optimism", "anger"]
    }
}
```

Le corps de réponse de l'API inclut le contenu suivant.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Décrire un volant

Utilisez l'DescribeFlywheelopération Amazon Comprehend pour récupérer les informations configurées concernant un volant d'inertie.

```
aws comprehend describe-flywheel \
    --flywheel-arn  "flywheelArn"
```

Le corps de réponse de l'API inclut le contenu suivant.

```
{
  "FlywheelProperties": {
      "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/
myTestFlywheel",
      "DataAccessRoleArn": "arn:aws::iam::111122223333:role/Admin",
      "TaskConfig": {
          "LanguageCode": "en",
          "DocumentClassificationConfig": {
              "Mode": "MULTI_LABEL"
          }
      },
      "DataLakeS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/
schemaVersion=1/20220801T014326Z",
      "Status": "ACTIVE",
      "ModelType":  "DOCUMENT_CLASSIFIER",
      "CreationTime": 1659318206.102,
      "LastModifiedTime": 1659318249.05
  }
}
```

Mettre à jour un volant

Utilisez cette UpdateFlywheelopération pour mettre à jour les valeurs de configuration modifiables du volant.

Certains champs de configuration sont des structures JSON avec des sous-champs. Pour mettre à jour un ou plusieurs sous-champs, fournissez des valeurs pour tous les sous-champs (Amazon Comprehend définit la valeur sur null pour tout sous-champ manquant dans la demande).

Si vous omettez un paramètre de premier niveau dans la `UpdateFlywheel` demande, Amazon Comprehend ne modifie pas les valeurs du paramètre ou de ses sous-champs dans le volant.

Pour ajouter ou supprimer des tags sur le volant, utilisez les UntagResourceopérations TagResourceet.

Vous pouvez promouvoir une version de modèle en définissant le `ActiveModelArn` paramètre, comme illustré dans l'exemple suivant.

```
aws comprehend update-flywheel \
    --region aws-region \
    --flywheel-arn  "flywheelArn" \
    --active-model-arn  "modelArn" \
```

Le corps de réponse de l'API inclut le contenu suivant.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Supprimer un volant

Utilisez l'DeleteFlywheelopération Amazon Comprehend pour supprimer les volants.

```
aws comprehend delete-flywheel \
    --flywheel-arn  "flywheelArn"
```

Une réponse API réussie contient un corps de message de réponse vide

Listez les volants

Utilisez l'ListFlywheelsopération Amazon Comprehend pour récupérer la liste des volants de la région actuelle.

```
aws comprehend list-flywheel \
    --region aws-region \
    --endpoint-url  "uri"
```

Le corps de réponse de l'API inclut le contenu suivant.

```
{
    "FlywheelSummaryList": [
        {
            "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/
myTestFlywheel",
            "DataLakeS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/
schemaVersion=1/20220801T014326Z",
            "Status": "ACTIVE",
            ""ModelType":  "DOCUMENT_CLASSIFIER",
            "CreationTime": 1659318206.102,
            "LastModifiedTime": 1659318249.05
        }
    ]
}
```

Configuration des ensembles de données

Pour ajouter des données d'entraînement ou de test étiquetées à un volant, utilisez la console ou l'API Amazon Comprehend pour créer un ensemble de données.

Vous configurez chaque ensemble de données en tant que données d'entraînement ou données de test. Vous associez le jeu de données à un volant spécifique et à un modèle personnalisé. Lorsque vous créez un ensemble de données, Amazon Comprehend télécharge les données dans le lac de données du volant. Pour plus de détails sur les formats de fichier pour les données d'entraînement, voir Préparation des données d'entraînement du classificateur ouPréparation des données de formation du système de reconnaissance d'entités.

Lorsque vous supprimez le volant, Amazon Comprehend supprime les ensembles de données. Les données téléchargées restent disponibles dans le lac de données.

Création d'un jeu de données (console)

Créer un jeu de données

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Volants et choisissez le volant dans lequel vous souhaitez ajouter les données.
3. Choisissez l'onglet Ensembles de données.
4. Dans le tableau Ensembles de données d'entraînement ou Ensembles de données de test, sélectionnez Créer un ensemble de données.
5. Sous Détails du jeu de données, entrez un nom pour le jeu de données et une description facultative.
6. Sous Spécifications des données, choisissez les champs de configuration du format des données et du type d'ensemble de données.
7. (Facultatif) Sous Format d'entrée, choisissez le format des documents d'entrée.
8. Sous Emplacement des annotations sur S3, entrez l'emplacement du fichier d'annotations sur Amazon S3.
9. Sous Emplacement des données d'entraînement sur S3, entrez l'emplacement des fichiers de documents sur Amazon S3.
10. Sélectionnez Create (Créer).

Création d'un jeu de données (API)

Vous pouvez utiliser cette CreateDatasetopération pour créer un ensemble de données.

Example

```
aws comprehend create-dataset \
    --flywheel-arn "myFlywheel2" \
    --dataset-name "my-training-dataset"
    --dataset-type "TRAIN"
    --description "my training dataset"
    --cli-input-json file://inputConfig.json
}
```

Le fichier `inputConfig.json` contient le contenu suivant.

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://my-comprehend-datasets/multilabel_train.csv"
    }
}
```

Pour ajouter ou supprimer des balises dans le jeu de données, utilisez les UntagResourceopérations TagResourceet.

Décrire un ensemble de données

Utilisez l'DescribeDatasetopération Amazon Comprehend pour récupérer les informations configurées concernant un volant d'inertie.

```
aws comprehend describe-dataset \
    --dataset-arn  "datasetARN"
```

La réponse contient le contenu suivant.

```
{
   "DatasetProperties": {
      "DatasetArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/
myTestFlywheel/dataset/train-dataset",
      "DatasetName": "train-dataset",
      "DatasetType": "TRAIN",
      "DatasetS3Uri": "s3://my-test-datalake/flywheelbasictest/myTestFlywheel/
schemaVersion=1/20220801T014326Z/datasets/train-dataset/20220801T194844Z",
      "Description": "Good Dataset",
      "Status": "COMPLETED",
      "NumberOfDocuments": 90,
      "CreationTime": 1659383324.297
  }
}
```

Itérations du volant

Utilisez les itérations du volant pour créer et gérer de nouvelles versions de modèles.

Rubriques

- Flux de travail d'itération
- Gestion des itérations (console)
- Gestion des itérations (API)

Flux de travail d'itération

Un volant démarre avec une version de modèle entraînée ou utilise un ensemble de données initial pour entraîner une version de modèle.

Au fil du temps, à mesure que vous obtenez de nouvelles données étiquetées, vous entraînez de nouvelles versions de modèles afin d'améliorer les performances de votre modèle de volant. Lorsque vous lancez le volant, il crée une nouvelle itération qui entraîne et évalue une nouvelle version du modèle. Vous pouvez promouvoir la nouvelle version du modèle si ses performances sont supérieures à celles de la version active existante.

Le flux de travail d'itération du volant comprend les étapes suivantes :

1. Vous créez des ensembles de données pour les nouvelles données étiquetées.
2. Vous lancez le volant pour créer une nouvelle itération. L'itération suit les étapes suivantes pour entraîner et évaluer une nouvelle version du modèle :
   a. Évalue la version active du modèle à l'aide des nouvelles données.
   b. Entraîne une nouvelle version du modèle en utilisant les nouvelles données.
   c. Stocke les résultats de l'évaluation et de la formation dans le lac de données.
   d. Renvoie les scores F1 pour les deux modèles.
3. Une fois l'itération terminée, vous pouvez comparer les scores F1 du modèle actif existant et du nouveau modèle.
4. Si la nouvelle version du modèle présente des performances supérieures, vous la promouvez en tant que version active du modèle. Vous pouvez utiliser la console ou l'API pour promouvoir la nouvelle version du modèle.

Gestion des itérations (console)

Vous pouvez utiliser la console pour démarrer une nouvelle itération et demander le statut d'une itération en cours. Vous pouvez également consulter les résultats des itérations terminées.

Lancer une itération sur le volant (console)

Avant de commencer une nouvelle itération, créez un ou plusieurs nouveaux ensembles de données d'entraînement ou de test. Consultez Configuration des ensembles de données.

Lancer une itération sur le volant (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Flywheels.
3. Dans le tableau Flywheels, choisissez un volant.
4. Choisissez Run flywheel.

Analyser les résultats des itérations (console)

Après avoir exécuté l'itération du volant, la console affiche les résultats dans le tableau des itérations des volants.

Promouvoir la nouvelle version du modèle (console)

Sur la page des détails du modèle de la console, vous pouvez promouvoir une nouvelle version du modèle en tant que version active du modèle.

Promouvoir une version de modèle de volant en version de modèle active (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la.
2. Dans le menu de gauche, choisissez Flywheels.
3. Dans le tableau Flywheels, choisissez un volant.
4. Dans le tableau de la page de détails du Flywheel, choisissez la version à promouvoir dans le tableau des itérations de Flywheels.
5. Choisissez Rendre le modèle actif.

Gestion des itérations (API)

Vous pouvez utiliser l'API Amazon Comprehend pour démarrer une nouvelle itération et demander le statut d'une itération en cours. Vous pouvez également consulter les résultats des itérations terminées.

Lancer l'itération du volant (API)

Utilisez l'StartFlywheelIterationopération Amazon Comprehend pour démarrer une itération en volant.

```
aws comprehend start-flywheel-iteration \
    --flywheel-arn  "flywheelArn"
```

La réponse contient le contenu suivant.

```
{
  "FlywheelIterationArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name"
}
```

Promouvoir la nouvelle version du modèle (API)

Utilisez cette UpdateFlywheelopération pour promouvoir une version de modèle en tant que version de modèle active.

Envoyez la `UpdateFlywheel` demande avec le `ActiveModelArn` paramètre défini à l'ARN de la nouvelle version du modèle actif.

```
aws comprehend update-flywheel \
    --active-model-arn  "modelArn" \
```

La réponse contient le contenu suivant.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Décrire les résultats de l'itération du volant (API)

L'DescribeFlywheelIterationopération Amazon Comprehend renvoie des informations sur une itération une fois celle-ci terminée.

```
aws comprehend describe-flywheel-iteration \
 --flywheel-arn "flywheelArn" \
 --flywheel-iteration-id  "flywheelIterationId" \
```

```
 --region aws-region
```

La réponse contient le contenu suivant.

```
{
    "FlywheelIterationProperties": {
        "FlywheelArn": "flywheelArn",
        "FlywheelIterationId": "iterationId",
        "CreationTime": <createdAt>,
        "EndTime": <endedAt>,
        "Status": <status>,
        "Message": <message>,
        "EvaluatedModelArn": "modelArn",
        "EvaluatedModelMetrics": {
            "AverageF1Score": <value>,
            "AveragePrecision": <value>,
            "AverageRecall": <value>,
            "AverageAccuracy": <value>
        },
        "TrainedModelArn": "modelArn",
        "TrainedModelMetrics": {
            "AverageF1Score": <value>,
            "AveragePrecision": <value>,
            "AverageRecall": <value>,
            "AverageAccuracy": <value>
        }
    }
}
```

Obtenir l'historique des itérations (API)

Utilisez cette ListFlywheelIterationHistoryopération pour obtenir des informations sur l'historique des itérations.

```
aws comprehend list-flywheel-iteration-history \
 --flywheel-arn "flywheelArn"
```

La réponse contient le contenu suivant.

```
{
    "FlywheelIterationPropertiesList": [
        {
```

```
            "FlywheelArn": "<flywheelArn>",
            "FlywheelIterationId": "20220907T214613Z",
            "CreationTime": 1662587173.224,
            "EndTime": 1662592043.02,
            "Status": "<status>",
            "Message": "<message>",
            "EvaluatedModelArn": "modelArn",
            "EvaluatedModelMetrics": {
                "AverageF1Score": 0.8333333333333333,
                "AveragePrecision": 0.75,
                "AverageRecall": 0.9375,
                "AverageAccuracy": 0.8125
            },
            "TrainedModelArn": "modelArn",
            "TrainedModelMetrics": {
                "AverageF1Score": 0.865497076023392,
                "AveragePrecision": 0.7636363636363637,
                "AverageRecall": 1.0,
                "AverageAccuracy": 0.84375
            }
        }
    ]
}
```

Utilisation de volants pour l'analyse

Vous pouvez utiliser la version active du modèle du volant pour exécuter une analyse en vue d'une classification personnalisée ou d'une reconnaissance d'entités. La version du modèle actif est configurable. Vous pouvez utiliser la console ou l'opération UpdateFlywheelAPI pour définir une nouvelle version du modèle comme version active du modèle.

Pour utiliser le volant, spécifiez l'ARN du volant plutôt qu'un ARN de modèle personnalisé lorsque vous configurez la tâche d'analyse. Amazon Comprehend exécute l'analyse à l'aide de la version du modèle actif du volant.

Analyse en temps réel

Vous utilisez un point de terminaison pour exécuter une analyse en temps réel. Lorsque vous créez ou mettez à jour un point de terminaison, vous pouvez le configurer avec l'ARN du volant au lieu d'un ARN du modèle. Lorsque vous exécutez l'analyse en temps réel, sélectionnez le point de terminaison

associé au volant. Amazon Comprehend exécute l'analyse à l'aide de la version du modèle actif du volant.

Lorsque vous définissez une nouvelle version du modèle actif pour le volant, le point de terminaison est automatiquement mis à jour pour commencer à utiliser la nouvelle version du modèle actif. UpdateFlywheel Si vous ne souhaitez pas que le point de terminaison soit mis à jour automatiquement, configurez-le (en utilisant UpdateEndpoint) pour qu'il utilise directement l'ARN de la version du modèle. Le terminal continue d'utiliser cette version du modèle si la version du modèle actif du volant change.

Pour une classification personnalisée, utilisez l'opération ClassifyDocumentAPI. Pour une reconnaissance d'entité personnalisée, utilisez la demande DetectEntitiesd'API. Indiquez l'extrémité du volant dans le EndpointArn paramètre.

Vous pouvez également utiliser la console pour exécuter une analyse en temps réel pour une classification personnalisée ou une reconnaissance d'entité personnalisée.

Tâches asynchrones

Pour une classification personnalisée, utilisez la demande StartDocumentClassificationJobd'API pour démarrer une tâche asynchrone. Fournissez le FlywheelArn paramètre au lieu duDocumentClassifierArn.

Pour une reconnaissance d'entité personnalisée, utilisez la demande StartEntitiesDetectionJobd'API. Fournissez le FlywheelArn paramètre au lieu duEntityRecognizerArn.

Vous pouvez utiliser la console pour exécuter des tâches d'analyse asynchrones à des fins de classification personnalisée ou de reconnaissance d'entités personnalisée. Lorsque vous créez la tâche, entrez l'ARN du volant dans le champ Modèle de reconnaissance ou de modèle de classificateur.

Gestion des points de terminaison Amazon Comprehend

Dans Amazon Comprehend, les points de terminaison mettent à disposition vos modèles personnalisés pour la classification en temps réel ou la détection d'entités. Après avoir créé un point de terminaison, vous pouvez y apporter des modifications en fonction de l'évolution des besoins de votre entreprise. Par exemple, vous pouvez surveiller l'utilisation de vos terminaux et appliquer le dimensionnement automatique pour configurer automatiquement le provisionnement des terminaux en fonction de vos besoins en matière de capacité. Vous pouvez gérer tous vos points de terminaison à partir d'une seule vue, et lorsque vous n'avez plus besoin d'un point de terminaison, vous pouvez le supprimer pour réduire les coûts.

Avant de pouvoir gérer un point de terminaison, vous devez en créer un. Pour de plus amples informations, consultez les procédures suivantes.

- Création d'un point de terminaison pour une classification personnalisée
- Création d'un point de terminaison pour la détection d'entités personnalisées

Rubriques

- Présentation des points de terminaison Amazon Comprehend
- Utilisation des points de terminaison Amazon Comprehend
- Surveillance des points de terminaison Amazon Comprehend
- Mise à jour des points de terminaison Amazon Comprehend
- Utilisation Trusted Advisor avec Amazon Comprehend
- Suppression des points de terminaison Amazon Comprehend
- Mise à l'échelle automatique avec les points de terminaison

Présentation des points de terminaison Amazon Comprehend

La page des points de terminaison de la console Amazon Comprehend vous fournit une vue globale de vos points de terminaison. Sur la page de présentation des terminaux, vous pouvez consulter tous vos terminaux en un seul endroit pour comprendre l'utilisation de vos terminaux par rapport à votre utilisation réelle des ressources. En haut à droite de la page des points de terminaison, vous pouvez spécifier les points de terminaison que vous souhaitez afficher : tous, les points de terminaison du classificateur personnalisé ou les points de terminaison de votre entité personnalisée.

Vous pouvez créer, mettre à jour, surveiller et supprimer des points de terminaison à partir de cette page. Dans la section de présentation des points de terminaison, vous pouvez consulter la liste de vos points de terminaison, les modèles personnalisés qu'ils hébergent, leur heure de création, le débit provisionné et l'état du point de terminaison. Lorsque vous sélectionnez un point de terminaison spécifique dans le tableau d'aperçu des points de terminaison, les détails du point de terminaison sont affichés.

De plus, si vous êtes client du AWS Business Support ou du AWS Enterprise Support, vous avez accès aux contrôles Trusted Advisor spécifiques à vos terminaux. Pour en savoir plus, consultez Utilisation Trusted Advisor avec Amazon Comprehend. Pour une liste complète des vérifications et des descriptions, consultez les meilleures pratiques de Trusted Advisor.

Pour plus d'informations sur la gestion de vos points de terminaison, consultez les rubriques suivantes.

- Utilisation des points de terminaison Amazon Comprehend
- Surveillance des points de terminaison Amazon Comprehend
- Mise à jour des points de terminaison Amazon Comprehend
- Utilisation Trusted Advisor avec Amazon Comprehend
- Suppression des points de terminaison Amazon Comprehend

> **Important**
>
> Le coût de la classification personnalisée en temps réel est basé à la fois sur le débit que vous définissez et sur la durée pendant laquelle le point de terminaison est actif. Si vous n'utilisez plus le terminal, ou si vous ne l'utilisez pas pendant une période prolongée, vous devez mettre en place une politique de dimensionnement automatique afin de réduire vos coûts. Ou, si vous n'utilisez plus de point de terminaison, vous pouvez le supprimer pour éviter d'encourir des frais supplémentaires. Pour de plus amples informations, veuillez consulter Mise à l'échelle automatique avec les points de terminaison.

Utilisation des points de terminaison Amazon Comprehend

Vous créez un point de terminaison pour exécuter une analyse en temps réel à l'aide d'un modèle personnalisé. Un point de terminaison inclut des ressources gérées qui rendent votre modèle personnalisé disponible pour une inférence en temps réel.

Amazon Comprehend attribue le débit à un point de terminaison à l'aide d'unités d'inférence (UI). Une interface utilisateur représente un débit de données de 100 caractères par seconde. Vous pouvez doter le terminal d'un maximum de 10 unités d'inférence. Vous pouvez augmenter ou diminuer le débit du point de terminaison en mettant à jour le point de terminaison.

Si vos documents d'entrée incluent des documents semi-structurés ou des fichiers image, le débit de 100 caractères par seconde correspond aux caractères extraits du fichier d'entrée. Le nombre de caractères IUs que vous attribuez à un point de terminaison dépend de la densité de caractères des documents d'entrée.

Les réponses de DetectEntitiesl'API ClassifyDocumentet incluent le nombre de caractères pour chaque page de saisie. Vous pouvez utiliser ces informations pour estimer le nombre d'unités d'inférence à provisionner pour atteindre le débit souhaité.

Une fois que vous avez terminé votre analyse en temps réel, supprimez le terminal car il est facturé tant qu'il est actif. Vous pouvez créer un autre point de terminaison lorsque vous êtes prêt à effectuer d'autres analyses en temps réel.

Pour plus d'informations sur le coût des terminaux, consultez Amazon Comprehend Pricing.

Après avoir créé un point de terminaison, vous pouvez le surveiller avec Amazon CloudWatch, le mettre à jour pour modifier ses unités d'inférence ou le supprimer lorsqu'il n'est plus nécessaire. Pour de plus amples informations, veuillez consulter Surveillance des points de terminaison Amazon Comprehend.

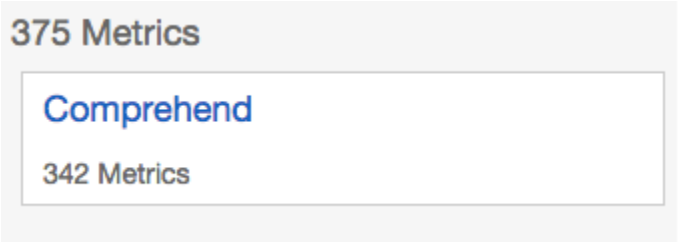
Surveillance des points de terminaison Amazon Comprehend

Vous pouvez ajuster le débit de votre point de terminaison en augmentant ou en diminuant le nombre d'unités d'inférence ()IUs. Pour plus d'informations sur la mise à jour de votre terminal, consultezthe section called “Mise à jour des terminaux”.

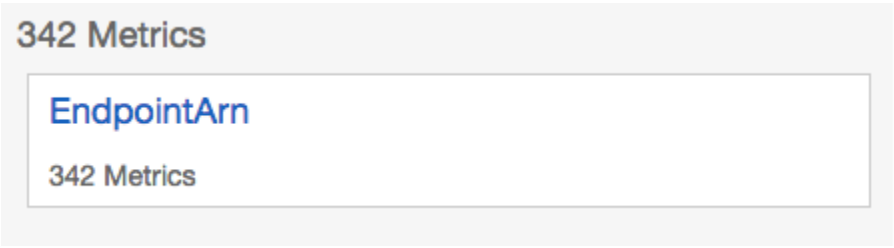
Vous pouvez déterminer la meilleure façon d'ajuster le débit de votre terminal en surveillant son utilisation à l'aide de la CloudWatch console Amazon.

Surveillez l'utilisation de vos terminaux avec CloudWatch

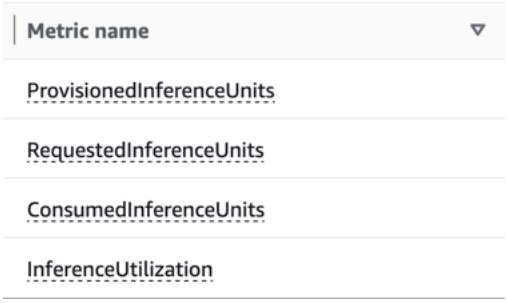
1. Connectez-vous à la CloudWatch console AWS Management Console et ouvrez-la.
2. Sur la gauche, choisissez Mesures, puis sélectionnez Toutes les mesures.
3. Sous Toutes les mesures, choisissez Comprehend.



4. La CloudWatch console affiche les dimensions des métriques Comprehend. Choisissez la EndpointArndimension.



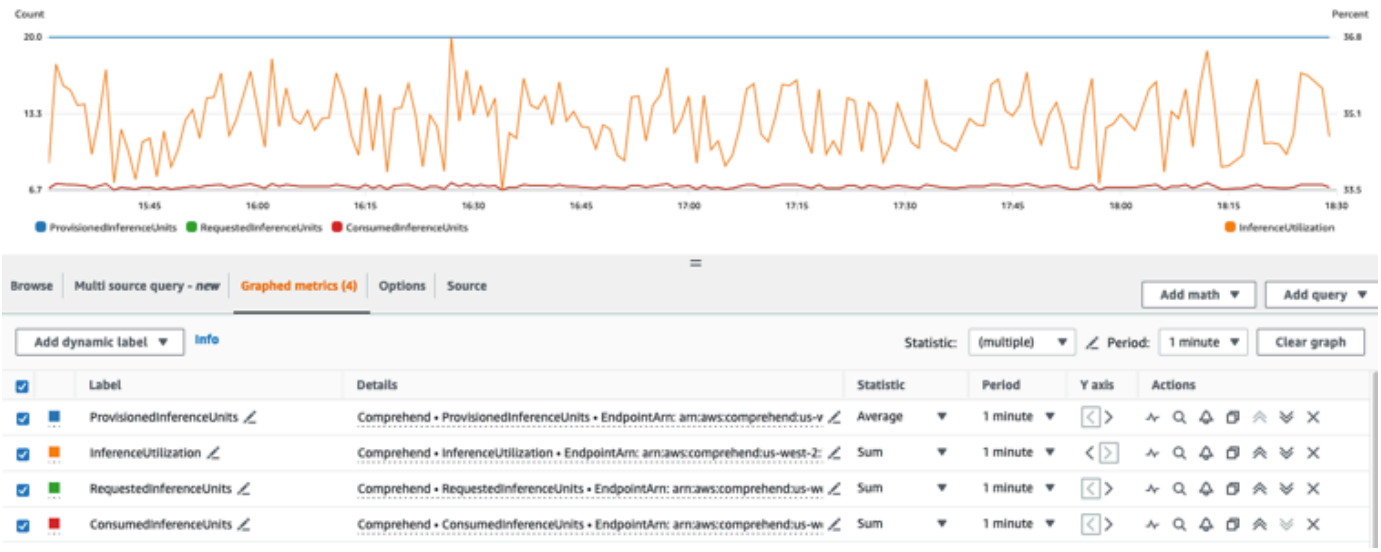
   La console affiche ProvisionedInferenceUnits, RequestedInferenceUnitsConsumedInferenceUnits, et InferenceUtilizationpour chacun de vos points de terminaison.



   Sélectionnez les quatre mesures et accédez à l'onglet Mesures graphiques.

5. Définissez les colonnes de statistiques pour RequestedInferenceUnitset ConsumedInferenceUnitspour Sum.
6. Définissez la colonne de statistiques sur InferenceUtilizationSomme.
7. Définissez la colonne des statistiques ProvisionedInferenceUnitssur Moyenne.
8. Modifiez la colonne Période pour toutes les mesures sur 1 minute.
9. Sélectionnez InferenceUtilizationet sélectionnez la flèche pour la déplacer vers un axe Y distinct.

   Votre graphique est prêt à être analysé.



Sur la base CloudWatch des indicateurs, vous pouvez également configurer le dimensionnement automatique pour ajuster automatiquement le débit de votre terminal. Pour plus d'informations sur l'utilisation de la mise à l'échelle automatique avec vos points de terminaison, consultezMise à l'échelle automatique avec les points de terminaison.

- ProvisionedInferenceUnits- Cette métrique représente le nombre moyen d'approvisionnements IUs au moment où la demande a été faite.
- RequestedInferenceUnits- Ceci est basé sur l'utilisation de chaque demande soumise au service qui a été envoyée pour être traitée. Cela peut être utile pour comparer la demande envoyée pour être traitée à ce qui a été réellement traité sans avoir recours à throttling ()ConsumedInferenceUnits. La valeur de cette métrique est calculée en divisant le nombre de caractères envoyés à traiter par le nombre de caractères pouvant être traités en une minute pour 1 UI.
- ConsumedInferenceUnits- Ceci est basé sur l'utilisation de chaque demande soumise au service qui a été traitée avec succès (non limitée). Cela peut être utile lorsque vous comparez ce que vous consommez à ce que vous avez provisionné IUs. La valeur de cette métrique est calculée en divisant le nombre de caractères traités par le nombre de caractères pouvant être traités en une minute pour 1 UI.
- InferenceUtilization- Ceci est émis par demande. Cette valeur est calculée en prenant la consommation IUs définie dans ConsumedInferenceUnits, en la divisant par ProvisionedInferenceUnitset en la convertissant en pourcentage sur 100.

> **Note**
>
> Toutes les métriques sont émises uniquement pour les demandes réussies. La métrique n'apparaîtra pas si elle provient d'une demande limitée ou si elle échoue en raison d'une erreur interne du serveur ou d'une erreur du client.

Mise à jour des points de terminaison Amazon Comprehend

Souvent, le niveau de débit dont vous avez besoin change après la création d'un point de terminaison ou après votre première estimation de vos besoins. Dans ce cas, il peut être nécessaire de mettre à jour votre terminal pour ajuster le débit à la hausse ou à la baisse. Le débit est régi par le nombre d'unités d'inférence avec lesquelles vous avez approvisionné votre terminal. Chaque unité d'inférence représente un débit de 100 caractères par seconde pour un maximum de 2 documents par seconde. Vous souhaiterez peut-être également mettre à jour la version du modèle associé au point de terminaison. Lorsque vous modifiez un point de terminaison, vous pouvez choisir une version différente du modèle pour le point de terminaison.

Il peut également être utile d'ajouter des balises à votre point de terminaison pour le garder organisé. Cela peut également être fait lors de la mise à jour de votre terminal. Pour plus d'informations sur les points de terminaison, voir Balisage de vos ressources

Pour mettre à jour un point de terminaison (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Endpoints.
3. Dans la liste des classificateurs, choisissez le nom du modèle personnalisé à partir duquel vous souhaitez mettre à jour le point de terminaison et suivez le lien. La page de détails du modèle s'affiche.
4. Sur la page des détails du modèle, sélectionnez les détails de la version. La liste des points de terminaison s'affiche.
5. Cochez la case correspondant à votre point de terminaison. En haut à droite du tableau des points de terminaison, sélectionnez l'icône Actions.
6. Choisissez Modifier. Vous pouvez mettre à jour les balises provisionnées IUs et modifier les balises.

7. Enregistrez vos modifications.
8. Pour modifier le nombre d'unités d'inférence avec lesquelles le point de terminaison est approvisionné, choisissez Modifier.
9. Entrez le nombre actualisé d'unités d'inférence à attribuer au point de terminaison. Chaque unité représente un débit de 100 caractères par seconde. Vous pouvez attribuer jusqu'à 10 unités d'inférence par point de terminaison.

   > **Note**
   >
   > Le coût d'utilisation d'un point de terminaison dépend de la durée de fonctionnement et du débit (basé sur le nombre d'unités d'inférence). L'augmentation du nombre d'unités d'inférence augmentera donc le coût de fonctionnement. Pour plus d'informations, consultez les tarifs d'Amazon Comprehend.

10. Choisissez Modifier le point de terminaison. La page de détails du point de terminaison s'affiche.
11. Vérifiez que le point de terminaison est en cours de mise à jour en choisissant le nom du modèle dans le fil d'Ariane en haut de la page. Sur la page des détails du modèle personnalisé, accédez à la liste des points de terminaison et vérifiez que l'option Mise à jour apparaît à côté du point de terminaison. Une fois la mise à jour terminée, le message « Prêt » s'affiche.

    L'exemple suivant illustre l'utilisation de l'UpdateEndpointopération avec la AWS CLI.

    L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

    ```
    aws comprehend update-endpoint \
        --desired-inference-units updated number of inference units \
        --desired-model-arn arn:aws:comprehend:region:account-id:model type/model name
     \
        --desired-data-access-role-arn arn:aws:iam:account id:role/role name
        --endpoint-arn arn:aws:comprehend:region:account id:endpoint/endpoint name
    ```

    Si l'action aboutit, Amazon Comprehend répond par une réponse HTTP 200 avec un corps HTTP vide.
12. Pour modifier le modèle personnalisé attaché à votre point de terminaison, depuis la page de détails du modèle personnalisé, accédez à la liste des points de terminaison.

13. Sélectionnez le point de terminaison que vous souhaitez modifier, puis sélectionnez Modifier.
14. Sur la page des paramètres du point de terminaison, sous Sélectionner un modèle de classificateur ou Sélectionner un modèle de reconnaissance en fonction de votre point de terminaison, vous pouvez rechercher un modèle dans la liste déroulante. Sélectionnez le modèle que vous souhaitez.
15. Sous Sélectionner une version, vous pouvez rechercher la version du modèle que vous souhaitez. Sélectionnez la version.
16. Sélectionnez Modifier le point de terminaison pour enregistrer.

Utilisation Trusted Advisor avec Amazon Comprehend

AWS Trusted Advisor est un outil en ligne qui fournit des recommandations pour vous aider à fournir vos ressources conformément aux AWS meilleures pratiques.

Si vous disposez d'un plan Basic ou d'un plan Support pour les développeurs, vous pouvez utiliser la Trusted Advisor console pour accéder à tous les contrôles de la catégorie Limites de service et à six contrôles de la catégorie Sécurité. Si vous avez un plan de Support Business ou Enterprise, vous pouvez utiliser la Trusted Advisor console et l'AWS Support API pour accéder à toutes les Trusted Advisor vérifications.

Amazon Comprehend prend en charge les Trusted Advisor contrôles suivants pour aider les clients à optimiser le coût et la sécurité de leurs points de terminaison Amazon Comprehend en fournissant des recommandations pratiques.

Points de terminaison sous-utilisés d'Amazon Comprehend

Le contrôle des points de terminaison sous-utilisés d'Amazon Comprehend évalue la configuration du débit de vos points de terminaison. Cette vérification vous avertit lorsque les points de terminaison ne sont pas activement utilisés pour les demandes d'inférence en temps réel. Un terminal qui n'est pas utilisé pendant plus de 15 jours est considéré comme sous-utilisé. Tous les terminaux sont facturés en fonction du débit défini et de la durée pendant laquelle le point de terminaison est actif. Pour le point de terminaison non utilisé au cours des 15 derniers jours, nous vous recommandons de définir une politique de dimensionnement pour la ressource à l'aide de l'outil Application Autoscaling. Pour un point de terminaison qui n'a pas été utilisé au cours des 30 derniers jours et pour lequel une politique de dimensionnement automatique est définie, nous vous recommandons d'utiliser l'inférence asynchrone ou de la supprimer. Les résultats de ces vérifications sont automatiquement actualisés

une fois par jour et peuvent être consultés dans la CostOptimizationcatégorie correspondante sur la Trusted Advisor console.

Pour consulter l'état d'utilisation de tous vos terminaux et les recommandations correspondantes

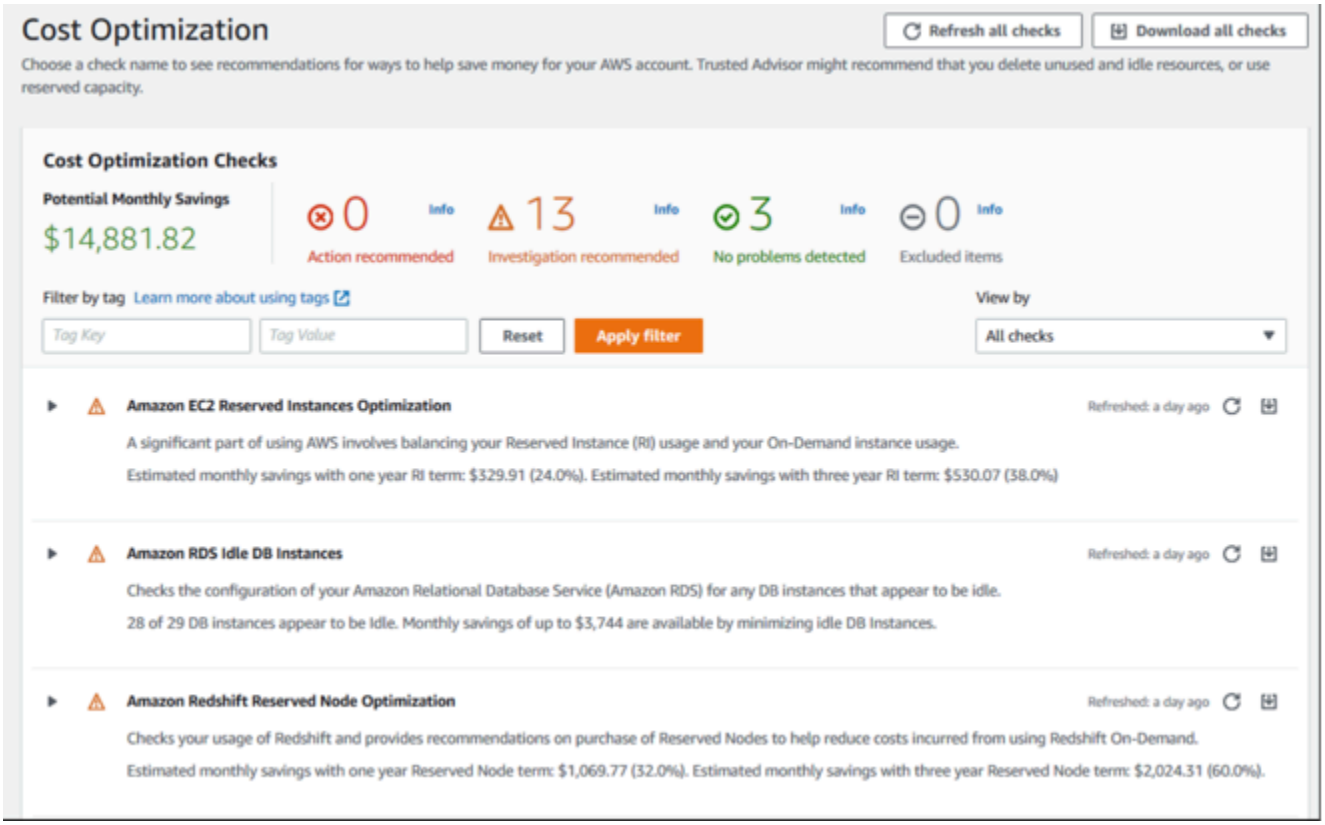
1. Connectez-vous à la Trusted Advisor console AWS Management Console et ouvrez-la.
2. Dans le volet de navigation, choisissez la catégorie de CostOptimizationvérification.
3. Sur la page des catégories, vous pouvez consulter le résumé de chaque catégorie de contrôle :
   - Action recommandée (rouge) : Trusted Advisor recommande une action pour le contrôle.
   - Investigation recommandée (jaune) : Trusted Advisor détecte un problème possible pour la vérification.
   - Aucun problème détecté (vert) : Trusted Advisor ne détecte aucun problème lors de la vérification.
   - Éléments exclus (gris) : nombre de contrôles comportant des éléments exclus, tels que les ressources que vous souhaitez ignorer lors d'un contrôle.
4. Choisissez le check Amazon Comprehend Underused Endpoints pour consulter la description du contrôle et les informations suivantes :
   - Alert Criteria (Critères d'alerte) : décrit le seuil à partir duquel une vérification va changer d'état.
   - Recommended Action (Action recommandée) : décrit les actions recommandées pour cette vérification.
   - Tableau des ressources : tableau répertoriant les détails de votre point de terminaison et le statut de chacun en fonction de vos recommandations.
5. Dans le tableau des ressources, si un point de terminaison est marqué par une enquête recommandée en raison d'un avertissement « Non utilisé au cours des 30 derniers jours », vous pouvez accéder à la page Détails du point de terminaison sur la console Amazon Comprehend.
   - Si vous ne souhaitez plus utiliser ce point de terminaison, choisissez Supprimer.
   - Choisissez Supprimer de nouveau pour confirmer la suppression. La page de détails du modèle personnalisé s'affiche. Vérifiez que le point de terminaison que vous avez supprimé est affiché à côté de celui-ci. Une fois supprimé, le point de terminaison est retiré de la liste des points de terminaison.
6. Dans le tableau des ressources de la Trusted Advisor console, si un point de terminaison est marqué avec le statut Investigation Recommended parce qu'il n'a pas été utilisé au cours des 15

derniers jours, et s'il est AutoScaling désactivé, vous pouvez accéder à la page Détails du point de terminaison sur la console Amazon Comprehend pour ajuster le point de terminaison.

- Si vous souhaitez réduire le débit configuré pour ce point de terminaison, cliquez sur Modifier. Entrez le nombre actualisé d'unités d'inférence à attribuer au point de terminaison, puis cochez la case pour confirmer, puis choisissez Modifier le point de terminaison. Lorsque la mise à jour est terminée, le statut indique Prêt.
- Si vous souhaitez configurer automatiquement le provisionnement des terminaux sur votre point de terminaison au lieu d'ajuster manuellement la configuration du débit, nous vous recommandons d'utiliser Application Autoscaling.

7. Dans le tableau des ressources de la Trusted Advisor console, si un point de terminaison est marqué avec le statut Aucun problème détecté en raison de l'utilisation active, cela signifie que le point de terminaison est utilisé activement pour exécuter des demandes d'inférence en temps réel et qu'aucune action n'est recommandée.

Voici un exemple qui montre l'affichage des CostOptimization catégories sur la Trusted Advisor console :



Risque lié à l'accès aux terminaux Amazon Comprehend

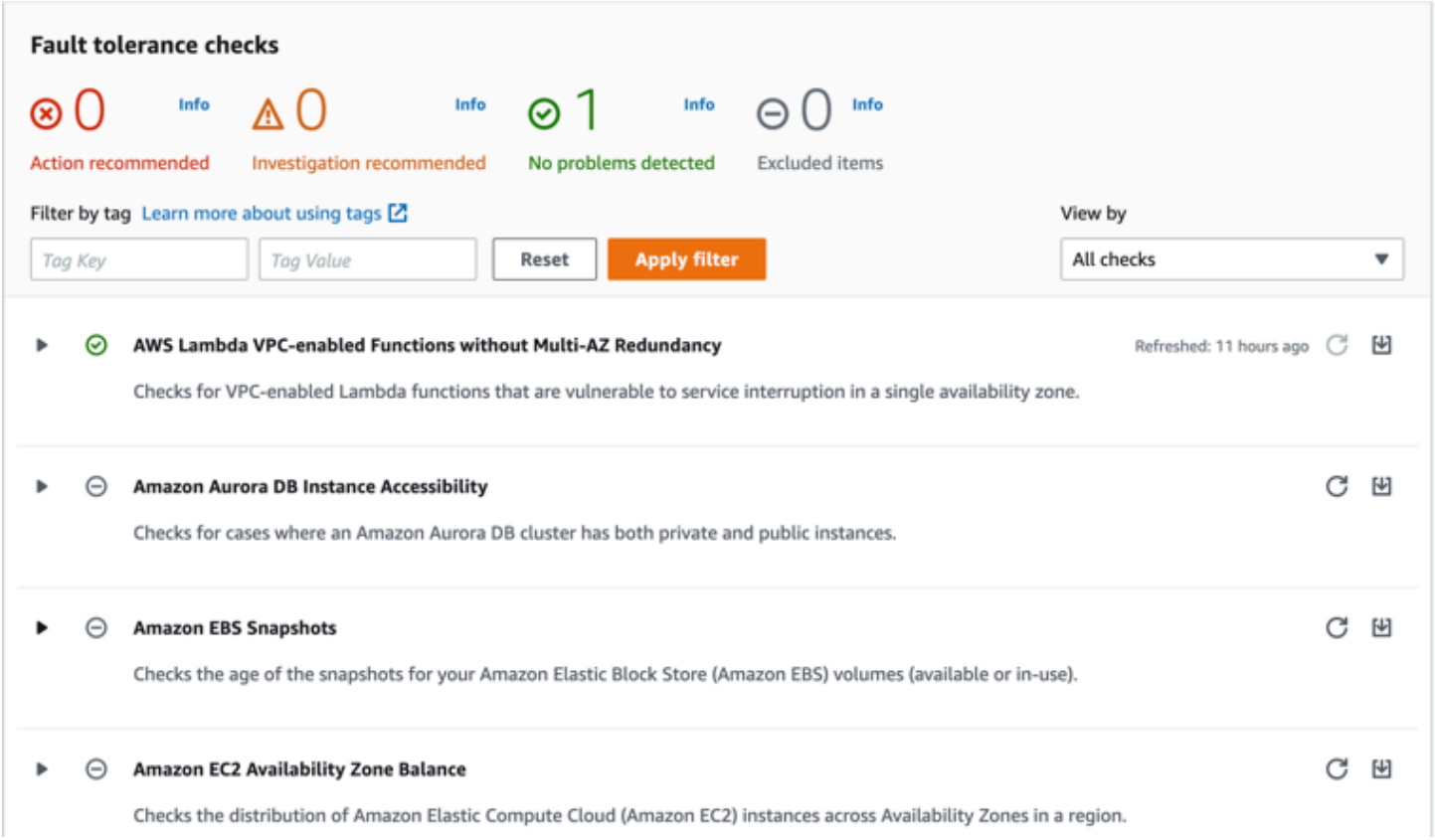
La vérification des risques d'accès aux terminaux Amazon Comprehend évalue les autorisations clés AWS Key Management Service (AWS KMS) pour un point de terminaison dont le modèle sous-jacent a été chiffré à l'aide de clés gérées par le client. Si la clé gérée par le client est désactivée ou si la politique en matière de clés a été modifiée pour modifier les autorisations autorisées pour Amazon Comprehend, la disponibilité du point de terminaison peut être affectée. Si la clé a été désactivée, nous vous recommandons de l'activer. Si la politique clé a été modifiée et que vous souhaitez continuer à utiliser ce point de terminaison, nous vous recommandons de mettre à jour la politique clé. Les résultats du contrôle sont automatiquement actualisés plusieurs fois par jour. Cette vérification peut être consultée dans la catégorie Fault Tolerance de la Trusted Advisor console.

Pour consulter le statut AWS KMS clé de vos points de terminaison Amazon Comprehend

1. Connectez-vous à la Trusted Advisor console AWS Management Console et ouvrez-la.
2. Dans le volet de navigation, choisissez la catégorie de FaultTolerancevérification.
3. Sur la page des catégories, vous pouvez consulter le résumé de chaque catégorie de contrôle :
   - Action recommandée (rouge) : Trusted Advisor recommande une action pour le contrôle.
   - Enquête recommandée (jaune) : Trusted Advisor détecte un éventuel problème lors de la vérification.
   - Aucun problème détecté (vert) : Trusted Advisor ne détecte aucun problème lors de la vérification.
   - Éléments exclus (gris) : nombre de vérifications qui ont exclu des éléments, tels que les ressources que vous souhaitez ignorer.
4. Choisissez Amazon Comprehend Endpoint Access Risk Check et vous pourrez consulter la description du contrôle et les informations suivantes :
   - Critères d'alerte — Décrit le seuil à partir duquel un contrôle changera de statut.
   - Recommended Action (Action recommandée) : décrit les actions recommandées pour cette vérification.
   - Tableau des ressources : tableau répertoriant les détails de votre point de terminaison chiffré KMS et l'état de chacun d'entre eux en fonction des actions recommandées.
5. Dans le tableau des ressources, si un point de terminaison est marqué avec le statut Action recommandée, sélectionnez le lien dans la KeyId colonne KMS et vous serez redirigé vers la page AWS KMS clé correspondante.

- Pour activer une AWS KMS touche désactivée, choisissez Actions clés, puis sélectionnez Activer.
- Si le statut clé est répertorié comme Activé, mettez à jour la politique clé en choisissant Passer à l'affichage des politiques dans la section Stratégie clé. Modifiez le document de politique clé pour fournir les autorisations nécessaires à Amazon Comprehend, puis choisissez Enregistrer les modifications.

Voici un exemple de l'affichage des FaultTolerance catégories sur la Trusted Advisor console :



Ces contrôles et leurs résultats peuvent également être consultés en consultant la Trusted Advisor section de l' AWS Support API.

Pour en savoir plus sur la configuration des alarmes à l'aide de CloudWatch, voir : Création d' Trusted Advisor alarmes à l'aide de CloudWatch. Pour un ensemble complet de vérifications des Trusted Advisor meilleures pratiques, voir : liste de contrôle des AWS Trusted Advisor meilleures pratiques.

Suppression des points de terminaison Amazon Comprehend

Une fois que vous n'avez plus besoin de votre terminal, vous devez le supprimer afin de ne plus en supporter les coûts. Vous pouvez facilement créer un autre point de terminaison chaque fois que vous en avez besoin dans la section Points de terminaison.

Pour supprimer un point de terminaison (console)

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Dans le menu de gauche, choisissez Endpoints.
3. Dans le tableau Endpoints, recherchez le point de terminaison que vous souhaitez supprimer. Vous pouvez rechercher ou filtrer tous les points de terminaison pour trouver celui dont vous avez besoin.
4. Cochez la case du point de terminaison correspondant au point de terminaison que vous souhaitez supprimer. En haut à droite du tableau des points de terminaison, sélectionnez l'icône Actions.
5. Sélectionnez Delete (Supprimer).
6. Choisissez Supprimer de nouveau pour confirmer la suppression. La page des points de terminaison s'affiche. Vérifiez que le point de terminaison que vous avez supprimé est affiché à côté de celui-ci. Lorsqu'il est supprimé, le point de terminaison est retiré de la liste des points de terminaison.

Pour supprimer un point de terminaison (AWS CLI)

L'exemple suivant illustre l'utilisation de l'DeleteEndpointopération avec la AWS CLI.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws comprehend delete-endpoint \
    --endpoint-arn arn:aws:comprehend:region:account-id endpoint/endpoint name
```

Si l'action aboutit, Amazon Comprehend répond par une réponse HTTP 200 avec un corps HTTP vide.

Mise à l'échelle automatique avec les points de terminaison

Au lieu d'ajuster manuellement le nombre d'unités d'inférence fournies pour vos points de terminaison de classification de documents et vos points de terminaison de reconnaissance d'entités, vous pouvez utiliser le dimensionnement automatique pour configurer automatiquement le provisionnement des points de terminaison en fonction de vos besoins en capacité.

Il existe deux manières d'utiliser le dimensionnement automatique pour ajuster le nombre d'unités d'inférence configurées pour votre point de terminaison :

- Suivi de la cible: configurez le dimensionnement automatique pour ajuster le provisionnement des terminaux en fonction des besoins de capacité en fonction de l'utilisation.
- Mise à l'échelle planifiée: configurez le dimensionnement automatique pour ajuster le provisionnement des terminaux en fonction des besoins de capacité selon un calendrier spécifié.

Vous ne pouvez définir la mise à l'échelle automatique qu'avec le AWS Command Line Interface (AWS CLI). Pour plus d'informations sur le dimensionnement automatique, voir Qu'est-ce qu'Application Auto Scaling ?

Suivi de la cible

Grâce au suivi des cibles, vous pouvez ajuster le provisionnement des terminaux en fonction de vos besoins en matière de capacité en fonction de l'utilisation. Le nombre d'unités d'inférence s'ajuste automatiquement pour que la capacité utilisée se situe dans les limites d'un pourcentage cible de la capacité allouée. Vous pouvez utiliser le suivi des cibles pour faire face aux pics d'utilisation temporaires de vos points de terminaison de classification de documents et de vos points de terminaison de reconnaissance d'entités. Pour plus d'informations, consultez Politiques de suivi des cibles et d'échelonnement pour Application Auto Scaling.

> **Note**
>
> Les exemples suivants sont formatés pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Configuration du suivi des cibles

Pour configurer le suivi des cibles pour un point de terminaison, vous utilisez les commandes de la AWS CLI pour enregistrer une cible évolutive, puis créer une politique de dimensionnement. La cible évolutive définit les unités d'inférence comme la ressource utilisée pour ajuster le provisionnement des terminaux, et la politique de dimensionnement définit les métriques qui contrôlent le dimensionnement automatique de la capacité provisionnée.

Pour configurer le suivi des cibles

1. Inscription d'une cible évolutive. Les exemples suivants enregistrent une cible évolutive pour ajuster le provisionnement des terminaux avec une capacité minimale d'une unité d'inférence et une capacité maximale de 2 unités d'inférence.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling register-scalable-target \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits \
       --min-capacity 1 \
       --max-capacity 2
   ```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling register-scalable-target \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
   endpoint/name \
       --scalable-dimension comprehend:entity-recognizer-
   endpoint:DesiredInferenceUnits \
       --min-capacity 1 \
       --max-capacity 2
   ```

2. Pour vérifier l'enregistrement de la cible évolutive, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace comprehend \
    --resource-id endpoint ARN
```

3. Créez une configuration de suivi cible pour la politique de dimensionnement et enregistrez la configuration dans un fichier appelé`config.json`. Voici un exemple de configuration de suivi cible pour un point de terminaison de classification de documents qui vise à maintenir la InferenceUtilization métrique à 70 %.

```
{
  "TargetValue": 70,
  "CustomizedMetricSpecification": {
    "MetricName": "InferenceUtilization",
    "Namespace": "MyNamespace",
    "Dimensions": [
      {
        "Name": "EndpointArn",
        "Value": "arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name"
      }
    ],
    "Statistic": "Sum",
    "Unit": "Percent"
  }
}
```

Voici un exemple de point de terminaison de reconnaissance d'entités :

```
{
  "TargetValue": 70,
  "CustomizedMetricSpecification": {
    "MetricName": "InferenceUtilization",
    "Namespace": "MyNamespace",
    "Dimensions": [
      {
        "Name": "EndpointArn",
        "Value": "arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name"
      }
    ],
```

```
    "Statistic": "Sum",
    "Unit": "Percent"
  }
}
```

4. Créez une politique de dimensionnement. Les exemples suivants créent une politique de dimensionnement basée sur la configuration de suivi des cibles définie dans le `config.json` fichier.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling put-scaling-policy \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits \
       --policy-name TestPolicy \
       --policy-type TargetTrackingScaling \
       --target-tracking-scaling-policy-configuration file://config.json
   ```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling put-scaling-policy \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
   endpoint/name \
       --scalable-dimension comprehend:entity-recognizer-
   endpoint:DesiredInferenceUnits \
       --policy-name TestPolicy \
       --policy-type TargetTrackingScaling \
       --target-tracking-scaling-policy-configuration file://config.json
   ```

Considérations

Les considérations suivantes s'appliquent lors de l'utilisation du suivi des cibles avec les points de terminaison Comprehend :

- Les métriques du point de terminaison ne sont émises que pour les demandes réussies. Les statistiques ne s'afficheront pas pour les demandes limitées ou qui échouent en raison d'une erreur interne du serveur ou d'une erreur du client.
- Lorsque des points de données sont manquants, l'état de l' CloudWatch alarme de sauvegarde passe à`INSUFFICIENT_DATA` . Dans ce cas, Application Auto Scaling ne peut pas redimensionner votre point de terminaison.
- Les mathématiques métriques peuvent être utiles pour contourner cette limitation. Par exemple, pour utiliser une valeur de 0 lorsqu'aucune métrique n'est signalée, utilisez la `FILL(m1,0)` fonction where `m1` is the metric. Il est important de tester votre configuration pour vous assurer qu'elle se comporte comme prévu. Voir Création de politiques de suivi des cibles à l'aide de mathématiques métriques pour d'autres options.

Supprimer le suivi des cibles

Pour supprimer le suivi des cibles pour un point de terminaison, vous utilisez les commandes de la AWS CLI pour supprimer la politique de dimensionnement, puis désenregistrer la cible évolutive.

Pour supprimer le suivi des cibles

1. Supprimez la politique de dimensionnement. Les exemples suivants suppriment une politique de dimensionnement spécifiée.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling delete-scaling-policy \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits \
       --policy-name TestPolicy \
   ```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling delete-scaling-policy \
       --service-namespace comprehend \
   ```

```
    --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
    --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
    --policy-name TestPolicy
```

2. Désenregistrez la cible évolutive Les exemples suivants désenregistrent une cible évolutive spécifiée.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling deregister-scalable-target \
    --service-namespace comprehend \
    --resource-id arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name \
    --scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits
```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling deregister-scalable-target \
    --service-namespace comprehend \
    --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
    --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits
```

Mise à l’échelle planifiée

Grâce au dimensionnement planifié, vous pouvez ajuster le provisionnement des terminaux en fonction de vos besoins en capacité selon un calendrier précis. La mise à l'échelle planifiée ajuste automatiquement le nombre d'unités d'inférence en fonction des pics d'utilisation à des moments précis. Vous pouvez utiliser le dimensionnement planifié pour les points de terminaison de classification de documents et les points de terminaison de reconnaissance d'entités. Pour plus d'informations sur le dimensionnement planifié, consultez la section Scaling programmé pour Application Auto Scaling.

> **Note**
>
> Les exemples suivants sont formatés pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Configuration du dimensionnement planifié

Pour configurer le dimensionnement planifié pour un point de terminaison, vous utilisez les commandes de la AWS CLI pour enregistrer une cible évolutive, puis créer une action planifiée. La cible évolutive définit les unités d'inférence comme la ressource utilisée pour ajuster le provisionnement des terminaux, et l'action planifiée contrôle le dimensionnement automatique de la capacité allouée à des moments précis.

Pour configurer le dimensionnement planifié

1. Inscription d'une cible évolutive. Les exemples suivants enregistrent une cible évolutive pour ajuster le provisionnement des terminaux avec une capacité minimale d'une unité d'inférence et une capacité maximale de 2 unités d'inférence.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling register-scalable-target \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits \
       --min-capacity 1 \
       --max-capacity 2
   ```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling register-scalable-target \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
   endpoint/name \
   ```

```
    --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
    --min-capacity 1 \
    --max-capacity 2
```

2. Créez une action planifiée. Les exemples suivants créent une action planifiée pour ajuster automatiquement la capacité allouée chaque jour à 12 h 00 UTC avec un minimum de 2 unités d'inférence et un maximum de 5 unités d'inférence. Pour plus d'informations sur les expressions chronologiques et le dimensionnement planifié, consultez la section Expressions de planification.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling put-scheduled-action \
    --service-namespace comprehend \
    --resource-id arn:aws:comprehend:region:account-id:document-classifier-
endpoint/name \
    --scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
    --scheduled-action-name TestScheduledAction \
    --schedule "cron(0 12 * * ? *)" \
    --scalable-target-action MinCapacity=2,MaxCapacity=5
```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling put-scheduled-action \
    --service-namespace comprehend \
    --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
    --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits \
    --scheduled-action-name TestScheduledAction \
    --schedule "cron(0 12 * * ? *)" \
    --scalable-target-action MinCapacity=2,MaxCapacity=5
```

Suppression du dimensionnement planifié

Pour supprimer le dimensionnement planifié pour un point de terminaison, vous utilisez les commandes de la AWS CLI pour supprimer l'action planifiée, puis désenregistrer la cible évolutive.

Pour supprimer le dimensionnement planifié

1. Supprimez l'action planifiée. Les exemples suivants suppriment une action planifiée spécifiée.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling delete-scheduled-action \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits \
       --scheduled-action-name TestScheduledAction
   ```

   Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling delete-scheduled-action \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
   endpoint/name \
       --scalable-dimension comprehend:entity-recognizer-
   endpoint:DesiredInferenceUnits \
       --scheduled-action-name TestScheduledAction
   ```

2. Désenregistrez la cible évolutive Les exemples suivants désenregistrent une cible évolutive spécifiée.

   Pour un point de terminaison de classification de documents, utilisez la commande AWS CLI suivante :

   ```
   aws application-autoscaling deregister-scalable-target \
       --service-namespace comprehend \
       --resource-id arn:aws:comprehend:region:account-id:document-classifier-
   endpoint/name \
       --scalable-dimension comprehend:document-classifier-
   endpoint:DesiredInferenceUnits
   ```

Pour un point de terminaison de reconnaissance d'entités, utilisez la commande AWS CLI suivante :

```
aws application-autoscaling deregister-scalable-target \
    --service-namespace comprehend \
    --resource-id arn:aws:comprehend:region:account-id:entity-recognizer-
endpoint/name \
    --scalable-dimension comprehend:entity-recognizer-
endpoint:DesiredInferenceUnits
```

Balisage de vos ressources

Une balise est une paire clé-valeur que vous pouvez ajouter à une ressource Amazon Comprehend sous forme de métadonnées. Vous pouvez utiliser des balises sur les tâches d'analyse, les modèles de classification personnalisés, les modèles de reconnaissance d'entités personnalisés et les points de terminaison. Les balises ont deux fonctions principales : organiser vos ressources et fournir un contrôle d'accès basé sur des balises.

Pour organiser vos ressources à l'aide de balises, vous pouvez ajouter la clé de balise « Département » et les valeurs de balise « Ventes » ou « Juridique ». Vous pouvez ensuite rechercher et filtrer les ressources pertinentes pour le service juridique de votre entreprise.

Pour fournir un contrôle d'accès basé sur des balises, créez des politiques IAM avec des autorisations basées sur des balises. Une politique peut autoriser ou interdire une opération en fonction des balises fournies dans votre demande (balises de demande) ou des balises associées à la ressource que vous appelez (balises de ressource). Pour plus d'informations sur l'utilisation des balises avec IAM, consultez la section Contrôle de l'accès à l'aide de balises dans le Guide de l'utilisateur IAM.

Considérations relatives à l'utilisation des balises avec Amazon Comprehend :

- Vous pouvez ajouter jusqu'à 50 balises par ressource, et les balises peuvent être ajoutées au moment de la création de la ressource ou rétroactivement.
- Une clé de balise est un champ obligatoire, mais une valeur de balise est facultative.
- Les balises ne doivent pas nécessairement être uniques entre les ressources, mais une ressource donnée ne peut pas avoir de clés de balise dupliquées.
- Les clés et valeurs de balise sont sensibles à la casse.
- Une clé de balise peut comporter un maximum de 127 caractères ; une valeur de balise peut comporter un maximum de 255 caractères.
- Le préfixe `aws:` « » est réservé à AWS l'usage ; vous ne pouvez pas ajouter, modifier ou supprimer des balises dont la clé commence `aws:` par. Ces tags ne sont pas pris en compte dans votre tags-per-resource limite de 50.

> **Note**
>
> Si vous prévoyez d'utiliser votre schéma de balisage pour plusieurs AWS services et ressources, n'oubliez pas que d'autres services peuvent avoir des exigences différentes en matière de caractères autorisés.

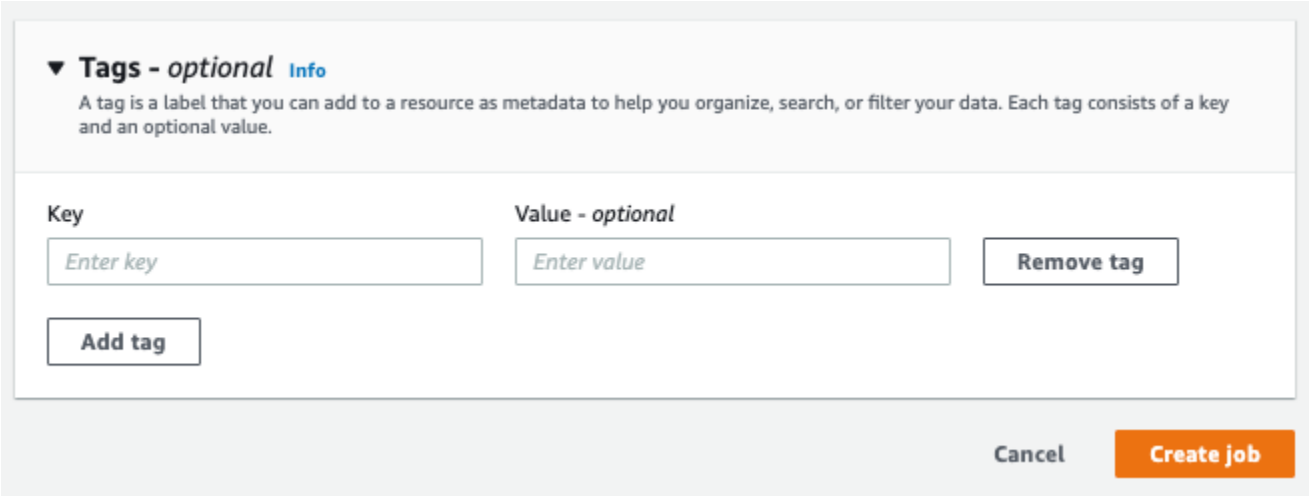
Rubriques

- Marquer une nouvelle ressource
- Afficher, modifier et supprimer les balises associées à une ressource

Marquer une nouvelle ressource

Vous pouvez ajouter des balises à une tâche d'analyse, à un modèle de classification personnalisé, à un modèle de reconnaissance d'entité personnalisé ou à des points de terminaison.

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Sélectionnez la ressource (tâche d'analyse, classification personnalisée ou reconnaissance d'entité personnalisée) que vous souhaitez créer dans le volet de navigation de gauche.
3. Cliquez sur Créer une tâche (ou sur Créer un nouveau modèle). Cela vous amène à la page principale de création de votre ressource. Au bas de cette page, vous verrez un panneau « Tags - facultatif ».



Saisissez une clé de balise et une valeur de balise facultative. Choisissez Ajouter un tag pour ajouter un autre tag à la ressource. Répétez ce processus jusqu'à ce que tous vos tags soient ajoutés. Notez que les clés de balise doivent être uniques par ressource.

4. Cliquez sur le bouton Créer ou Créer une tâche pour continuer à créer votre ressource.

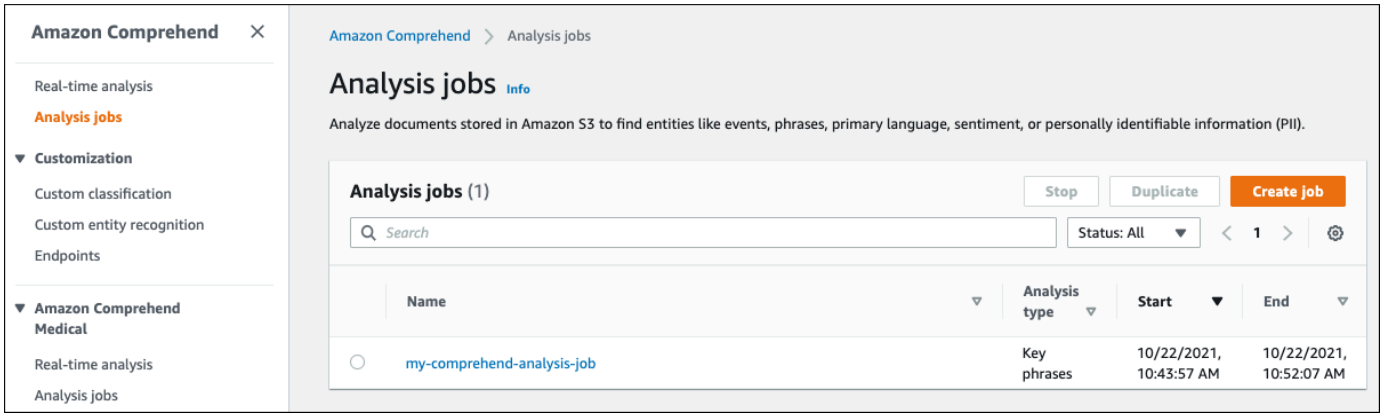
Vous pouvez également ajouter des balises à l'aide de la AWS CLI. Cet exemple montre comment ajouter des balises à l'aide de la start-entities-detection-jobcommande.

```
aws comprehend start-entities-detection-job \
--language-code "en" \
--input-data-config "{\"S3Uri\": \"s3://test-input/TEST.csv\"}" \
--output-data-config "{\"S3Uri\": \"s3://test-output\"}" \
--data-access-role-arn arn:aws:iam::123456789012:role/test \
--tags "[{\"Key\": \"color\",\"Value\": \"orange\"}]"
```

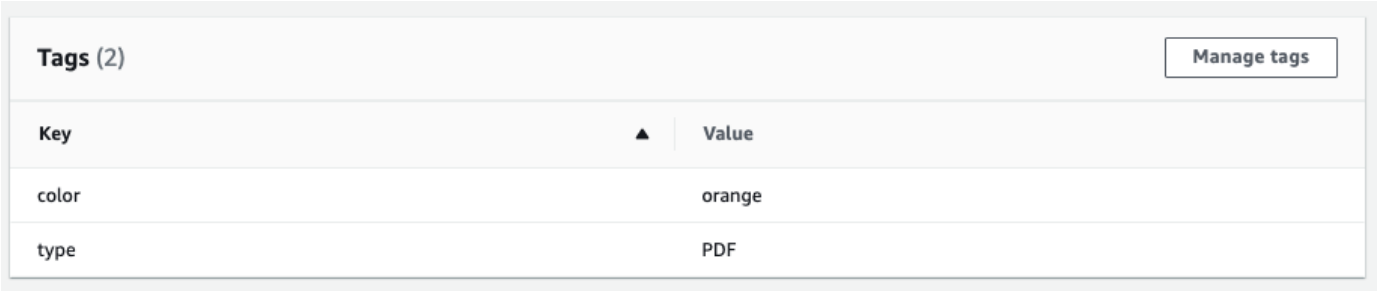
Afficher, modifier et supprimer les balises associées à une ressource

Vous pouvez afficher les balises associées à une tâche d'analyse, à un modèle de classification personnalisé ou à un modèle de reconnaissance d'entité personnalisé.

1. Connectez-vous à la console Amazon Comprehend AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/comprehend/
2. Sélectionnez la ressource (tâche d'analyse, classification personnalisée ou reconnaissance d'entité personnalisée) qui contient le fichier contenant les balises que vous souhaitez afficher, modifier ou supprimer. Cela affiche la liste des fichiers existants pour la ressource sélectionnée.

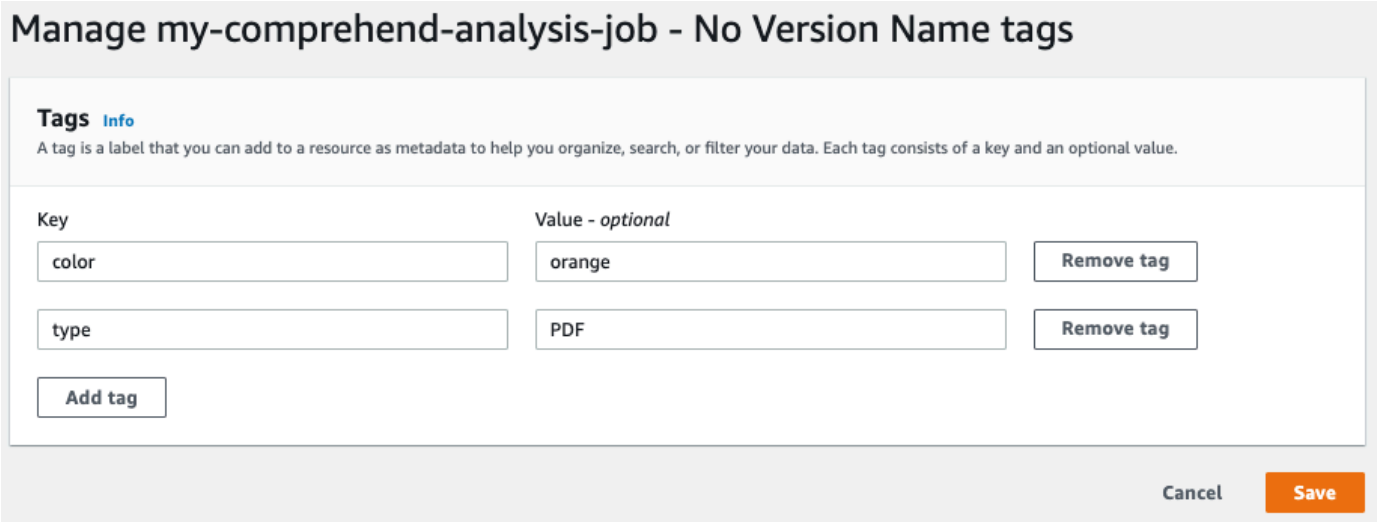


3. Cliquez sur le nom du fichier (ou du modèle) dont vous souhaitez afficher, modifier ou supprimer les balises. Cela vous amène à la page de détails de ce fichier (ou modèle). Faites défiler l'écran vers le bas jusqu'à ce que vous voyiez une boîte de dialogue Tags Vous pouvez voir ici toutes les balises associées au fichier (ou modèle) sélectionné.



Sélectionnez Gérer les balises pour modifier ou supprimer les balises de votre ressource.

4. Cliquez sur le texte que vous souhaitez modifier, puis modifiez votre tag. Vous pouvez également supprimer le tag en sélectionnant Supprimer le tag. Pour ajouter un nouveau tag, sélectionnez Ajouter un tag, puis saisissez le texte souhaité dans les champs vides.



Lorsque vous avez terminé de modifier vos balises, sélectionnez Enregistrer.

Exemples de code pour Amazon Comprehend à l'aide de AWS SDKs

Les exemples de code suivants montrent comment utiliser Amazon Comprehend avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de base pour l'utilisation d'Amazon Comprehend AWS SDKs
  - Actions utilisées par Amazon Comprehend AWS SDKs
    - Utilisation CreateDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DeleteDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DescribeDocumentClassificationJob avec un AWS SDK ou une CLI
    - Utilisation DescribeDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DescribeTopicsDetectionJob avec un AWS SDK ou une CLI
    - Utilisation DetectDominantLanguage avec un AWS SDK ou une CLI
    - Utilisation DetectEntities avec un AWS SDK ou une CLI
    - Utilisation DetectKeyPhrases avec un AWS SDK ou une CLI
    - Utilisation DetectPiiEntities avec un AWS SDK ou une CLI
    - Utilisation DetectSentiment avec un AWS SDK ou une CLI
    - Utilisation DetectSyntax avec un AWS SDK ou une CLI
    - Utilisation ListDocumentClassificationJobs avec un AWS SDK ou une CLI

    - Utilisation ListDocumentClassifiers avec un AWS SDK ou une CLI
    - Utilisation ListTopicsDetectionJobs avec un AWS SDK ou une CLI
    - Utilisation StartDocumentClassificationJob avec un AWS SDK ou une CLI
    - Utilisation StartTopicsDetectionJob avec un AWS SDK ou une CLI
- Scénarios d'utilisation d'Amazon Comprehend AWS SDKs
    - Créer une application de streaming Amazon Transcribe
    - Créez un chatbot Amazon Lex pour engager les visiteurs de votre site Web
    - Créez une application Web qui envoie et récupère des messages à l'aide d'Amazon SQS
    - Créez une application qui analyse les commentaires des clients et synthétise le son
    - Détectez les éléments d'un document avec Amazon Comprehend et un SDK AWS
    - Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK
    - Exécutez une tâche de modélisation de sujets Amazon Comprehend sur des exemples de données à l'aide d'un SDK AWS
    - Formez un classificateur Amazon Comprehend personnalisé et classez des documents à l'aide d'un SDK AWS

Exemples de base pour l'utilisation d'Amazon Comprehend AWS SDKs

Les exemples de code suivants montrent comment utiliser les bases d'Amazon Comprehend avec. AWS SDKs

Exemples

- Actions utilisées par Amazon Comprehend AWS SDKs
    - Utilisation CreateDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DeleteDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DescribeDocumentClassificationJob avec un AWS SDK ou une CLI
    - Utilisation DescribeDocumentClassifier avec un AWS SDK ou une CLI
    - Utilisation DescribeTopicsDetectionJob avec un AWS SDK ou une CLI
    - Utilisation DetectDominantLanguage avec un AWS SDK ou une CLI
    - Utilisation DetectEntities avec un AWS SDK ou une CLI
    - Utilisation DetectKeyPhrases avec un AWS SDK ou une CLI

- Utilisation DetectPiiEntities avec un AWS SDK ou une CLI
- Utilisation DetectSentiment avec un AWS SDK ou une CLI
- Utilisation DetectSyntax avec un AWS SDK ou une CLI
- Utilisation ListDocumentClassificationJobs avec un AWS SDK ou une CLI
- Utilisation ListDocumentClassifiers avec un AWS SDK ou une CLI
- Utilisation ListTopicsDetectionJobs avec un AWS SDK ou une CLI
- Utilisation StartDocumentClassificationJob avec un AWS SDK ou une CLI
- Utilisation StartTopicsDetectionJob avec un AWS SDK ou une CLI

Actions utilisées par Amazon Comprehend AWS SDKs

Les exemples de code suivants montrent comment effectuer des actions Amazon Comprehend individuelles avec. AWS SDKs Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Ces extraits appellent l'API Amazon Comprehend et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Vous pouvez voir les actions dans leur contexte dans Scénarios d'utilisation d'Amazon Comprehend AWS SDKs .

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour une liste complète, consultez le manuel Amazon Comprehend API Reference.

Exemples

- Utilisation CreateDocumentClassifier avec un AWS SDK ou une CLI
- Utilisation DeleteDocumentClassifier avec un AWS SDK ou une CLI
- Utilisation DescribeDocumentClassificationJob avec un AWS SDK ou une CLI
- Utilisation DescribeDocumentClassifier avec un AWS SDK ou une CLI
- Utilisation DescribeTopicsDetectionJob avec un AWS SDK ou une CLI
- Utilisation DetectDominantLanguage avec un AWS SDK ou une CLI
- Utilisation DetectEntities avec un AWS SDK ou une CLI
- Utilisation DetectKeyPhrases avec un AWS SDK ou une CLI
- Utilisation DetectPiiEntities avec un AWS SDK ou une CLI
- Utilisation DetectSentiment avec un AWS SDK ou une CLI
- Utilisation DetectSyntax avec un AWS SDK ou une CLI

- Utilisation ListDocumentClassificationJobs avec un AWS SDK ou une CLI
- Utilisation ListDocumentClassifiers avec un AWS SDK ou une CLI
- Utilisation ListTopicsDetectionJobs avec un AWS SDK ou une CLI
- Utilisation StartDocumentClassificationJob avec un AWS SDK ou une CLI
- Utilisation StartTopicsDetectionJob avec un AWS SDK ou une CLI

Utilisation **`CreateDocumentClassifier`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateDocumentClassifier`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour créer un classificateur de documents afin de classer les documents

L'`create-document-classifier`exemple suivant commence le processus de formation pour un modèle de classificateur de documents. Le fichier de données d'`training.csv`entraînement se trouve dans l'`--input-data-config`étiquette. `training.csv`est un document à deux colonnes où les étiquettes ou les classifications sont fournies dans la première colonne et les documents sont fournis dans la deuxième colonne.

```
aws comprehend create-document-classifier \
    --document-classifier-name example-classifier \
    --data-access-arn arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/123456abcdeb0e11022f22a11EXAMPLE \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --language-code en
```

Sortie :

```
{
```

```
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir CreateDocumentClassifierla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import
 software.amazon.awssdk.services.comprehend.model.CreateDocumentClassifierRequest;
import
 software.amazon.awssdk.services.comprehend.model.CreateDocumentClassifierResponse;
import
 software.amazon.awssdk.services.comprehend.model.DocumentClassifierInputDataConfig;

/**
 * Before running this code example, you can setup the necessary resources, such
 * as the CSV file and IAM Roles, by following this document:
 * https://aws.amazon.com/blogs/machine-learning/building-a-custom-classifier-
using-amazon-comprehend/
 *
 * Also, set up your development environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
```

```
 */
public class DocumentClassifierDemo {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <dataAccessRoleArn> <s3Uri> <documentClassifierName>

                Where:
                  dataAccessRoleArn - The ARN value of the role used for this
 operation.
                  s3Uri - The Amazon S3 bucket that contains the CSV file.
                  documentClassifierName - The name of the document classifier.
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String dataAccessRoleArn = args[0];
        String s3Uri = args[1];
        String documentClassifierName = args[2];

        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
                .region(region)
                .build();

        createDocumentClassifier(comClient, dataAccessRoleArn, s3Uri,
 documentClassifierName);
        comClient.close();
    }

    public static void createDocumentClassifier(ComprehendClient comClient,
 String dataAccessRoleArn, String s3Uri,
            String documentClassifierName) {
        try {
            DocumentClassifierInputDataConfig config =
 DocumentClassifierInputDataConfig.builder()
                    .s3Uri(s3Uri)
                    .build();

            CreateDocumentClassifierRequest createDocumentClassifierRequest =
 CreateDocumentClassifierRequest.builder()
```

```
                    .documentClassifierName(documentClassifierName)
                    .dataAccessRoleArn(dataAccessRoleArn)
                    .languageCode("en")
                    .inputDataConfig(config)
                    .build();

            CreateDocumentClassifierResponse createDocumentClassifierResult =
 comClient
                    .createDocumentClassifier(createDocumentClassifierRequest);
            String documentClassifierArn =
 createDocumentClassifierResult.documentClassifierArn();
            System.out.println("Document Classifier ARN: " +
 documentClassifierArn);

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateDocumentClassifierà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
```

```
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def create(
        self,
        name,
        language_code,
        training_bucket,
        training_key,
        data_access_role_arn,
        mode,
    ):
        """
        Creates a custom classifier. After the classifier is created, it
immediately
        starts training on the data found in the specified Amazon S3 bucket.
Training
        can take 30 minutes or longer. The `describe_document_classifier`
function
        can be used to get training status and returns a status of TRAINED when
the
        classifier is ready to use.

        :param name: The name of the classifier.
        :param language_code: The language the classifier can operate on.
        :param training_bucket: The Amazon S3 bucket that contains the training
data.
        :param training_key: The prefix used to find training data in the
training
                             bucket. If multiple objects have the same prefix,
all
                             of them are used.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                                     grants Comprehend permission to read from
the
                                     training bucket.
        :return: The ARN of the newly created classifier.
        """
        try:
            response = self.comprehend_client.create_document_classifier(
                DocumentClassifierName=name,
```

```
                LanguageCode=language_code,
                InputDataConfig={"S3Uri": f"s3://{training_bucket}/
{training_key}"},
                DataAccessRoleArn=data_access_role_arn,
                Mode=mode.value,
            )
            self.classifier_arn = response["DocumentClassifierArn"]
            logger.info("Started classifier creation. Arn is: %s.",
 self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't create classifier %s.", name)
            raise
        else:
            return self.classifier_arn
```

- Pour plus de détails sur l'API, consultez CreateDocumentClassifierle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteDocumentClassifier`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DeleteDocumentClassifier`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour supprimer un classificateur de documents personnalisé

L'`delete-document-classifier`exemple suivant supprime un modèle de classificateur de documents personnalisé.

```
aws comprehend delete-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section Gestion des points de terminaison Amazon Comprehend dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DeleteDocumentClassifierla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def delete(self):
        """
        Deletes the classifier.
        """
        try:
```

```
            self.comprehend_client.delete_document_classifier(
                DocumentClassifierArn=self.classifier_arn
            )
            logger.info("Deleted classifier %s.", self.classifier_arn)
            self.classifier_arn = None
        except ClientError:
            logger.exception("Couldn't deleted classifier %s.",
self.classifier_arn)
            raise
```

- Pour plus de détails sur l'API, consultez DeleteDocumentClassifierle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeDocumentClassificationJob`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeDocumentClassificationJob`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour décrire une tâche de classification de documents

L'`describe-document-classification-job`exemple suivant obtient les propriétés d'une tâche de classification de documents asynchrone.

```
aws comprehend describe-document-classification-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "DocumentClassificationJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "exampleclassificationjob",
        "JobStatus": "COMPLETED",
        "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
        "EndTime": "2023-06-14T17:15:58.582000+00:00",
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/mymodel/version/1",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-CLN-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole"
    }
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DescribeDocumentClassificationJobla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def describe_job(self, job_id):
        """
        Gets metadata about a classification job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
        """
        try:
            response =
 self.comprehend_client.describe_document_classification_job(
                JobId=job_id
            )
            job = response["DocumentClassificationJobProperties"]
            logger.info("Got classification job %s.", job["JobName"])
        except ClientError:
            logger.exception("Couldn't get classification job %s.", job_id)
            raise
        else:
            return job
```

- Pour plus de détails sur l'API, consultez DescribeDocumentClassificationJoble AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeDocumentClassifier`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeDocumentClassifier`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour décrire un classificateur de documents

L'`describe-document-classifier`exemple suivant obtient les propriétés d'un modèle de classificateur de documents personnalisé.

```
aws comprehend describe-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Sortie :

```
{
    "DocumentClassifierProperties": {
        "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/example-classifier-1",
        "LanguageCode": "en",
        "Status": "TRAINED",
        "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
        "EndTime": "2023-06-13T19:42:31.752000+00:00",
        "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
        "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
        "InputDataConfig": {
            "DataFormat": "COMPREHEND_CSV",
            "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
        },
        "OutputDataConfig": {},
        "ClassifierMetadata": {
```

```
            "NumberOfLabels": 3,
            "NumberOfTrainedDocuments": 5016,
            "NumberOfTestDocuments": 557,
            "EvaluationMetrics": {
                "Accuracy": 0.9856,
                "Precision": 0.9919,
                "Recall": 0.9459,
                "F1Score": 0.9673,
                "MicroPrecision": 0.9856,
                "MicroRecall": 0.9856,
                "MicroF1Score": 0.9856,
                "HammingLoss": 0.0144
            }
        },
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role",
        "Mode": "MULTI_CLASS"
    }
}
```

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DescribeDocumentClassifierla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
```

```
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def describe(self, classifier_arn=None):
        """
        Gets metadata about a custom classifier, including its current status.

        :param classifier_arn: The ARN of the classifier to look up.
        :return: Metadata about the classifier.
        """
        if classifier_arn is not None:
            self.classifier_arn = classifier_arn
        try:
            response = self.comprehend_client.describe_document_classifier(
                DocumentClassifierArn=self.classifier_arn
            )
            classifier = response["DocumentClassifierProperties"]
            logger.info("Got classifier %s.", self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't get classifier %s.", self.classifier_arn)
            raise
        else:
            return classifier
```

- Pour plus de détails sur l'API, consultez DescribeDocumentClassifierle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeTopicsDetectionJob`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeTopicsDetectionJob`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Exécuter une tâche de modélisation de rubrique sur des exemples de données

CLI

AWS CLI

Pour décrire une tâche de détection de sujets

L'`describe-topics-detection-job`exemple suivant obtient les propriétés d'une tâche de détection de sujets asynchrones.

```
aws comprehend describe-topics-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Sortie :

```
{
    "TopicsDetectionJobProperties": {
        "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
        "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
        "JobName": "example_topics_detection",
        "JobStatus": "IN_PROGRESS",
        "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
        "InputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-bucket",
            "InputFormat": "ONE_DOC_PER_LINE"
        },
        "OutputDataConfig": {
            "S3Uri": "s3://amzn-s3-demo-destination-bucket/
testfolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
        },
        "NumberOfTopics": 10,
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-examplerole"
    }
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DescribeTopicsDetectionJobla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def describe_job(self, job_id):
        """
        Gets metadata about a topic modeling job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
        """
        try:
            response = self.comprehend_client.describe_topics_detection_job(
                JobId=job_id
            )
            job = response["TopicsDetectionJobProperties"]
            logger.info("Got topic detection job %s.", job_id)
        except ClientError:
            logger.exception("Couldn't get topic detection job %s.", job_id)
            raise
        else:
            return job
```

- Pour plus de détails sur l'API, consultez DescribeTopicsDetectionJoble AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectDominantLanguage`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectDominantLanguage`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example calls the Amazon Comprehend service to determine the
/// dominant language.
/// </summary>
```

```
    public static class DetectDominantLanguage
    {
        /// <summary>
        /// Calls Amazon Comprehend to determine the dominant language used in
        /// the sample text.
        /// </summary>
        public static async Task Main()
        {
            string text = "It is raining today in Seattle.";

            var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

            Console.WriteLine("Calling DetectDominantLanguage\n");
            var detectDominantLanguageRequest = new
DetectDominantLanguageRequest()
            {
                Text = text,
            };

            var detectDominantLanguageResponse = await
comprehendClient.DetectDominantLanguageAsync(detectDominantLanguageRequest);
            foreach (var dl in detectDominantLanguageResponse.Languages)
            {
                Console.WriteLine($"Language Code: {dl.LanguageCode}, Score:
{dl.Score}");
            }

            Console.WriteLine("Done");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectDominantLanguageà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter la langue dominante du texte saisi

Ce qui suit `detect-dominant-language` analyse le texte saisi et identifie la langue dominante. Le score de confiance du modèle préentraîné est également affiché.

```
aws comprehend detect-dominant-language \
    --text "It is a beautiful day in Seattle."
```

Sortie :

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9877256155014038
        }
    ]
}
```

Pour plus d'informations, consultez Dominant Language dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DetectDominantLanguagela section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import
 software.amazon.awssdk.services.comprehend.model.DetectDominantLanguageRequest;
import
 software.amazon.awssdk.services.comprehend.model.DetectDominantLanguageResponse;
import software.amazon.awssdk.services.comprehend.model.DominantLanguage;
```

```
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetectLanguage {
    public static void main(String[] args) {
        // Specify French text - "It is raining today in Seattle".
        String text = "Il pleut aujourd'hui à Seattle";
        Region region = Region.US_EAST_1;

        ComprehendClient comClient = ComprehendClient.builder()
                .region(region)
                .build();

        System.out.println("Calling DetectDominantLanguage");
        detectTheDominantLanguage(comClient, text);
        comClient.close();
    }

    public static void detectTheDominantLanguage(ComprehendClient comClient,
 String text) {
        try {
            DetectDominantLanguageRequest request =
 DetectDominantLanguageRequest.builder()
                    .text(text)
                    .build();

            DetectDominantLanguageResponse resp =
 comClient.detectDominantLanguage(request);
            List<DominantLanguage> allLanList = resp.languages();
            for (DominantLanguage lang : allLanList) {
                System.out.println("Language is " + lang.languageCode());
            }

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectDominantLanguageà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_languages(self, text):
        """
        Detects languages used in a document.

        :param text: The document to inspect.
        :return: The list of languages along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_dominant_language(Text=text)
            languages = response["Languages"]
            logger.info("Detected %s languages.", len(languages))
        except ClientError:
            logger.exception("Couldn't detect languages.")
```

```
            raise
        else:
            return languages
```

- Pour plus de détails sur l'API, consultez DetectDominantLanguagele AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectEntities`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectEntities`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use the AmazonComprehend service detect any
```

```
    /// entities in submitted text.
    /// </summary>
    public static class DetectEntities
    {
        /// <summary>
        /// The main method calls the DetectEntitiesAsync method to find any
        /// entities in the sample code.
        /// </summary>
        public static async Task Main()
        {
            string text = "It is raining today in Seattle";

            var comprehendClient = new AmazonComprehendClient();

            Console.WriteLine("Calling DetectEntities\n");
            var detectEntitiesRequest = new DetectEntitiesRequest()
            {
                Text = text,
                LanguageCode = "en",
            };
            var detectEntitiesResponse = await
comprehendClient.DetectEntitiesAsync(detectEntitiesRequest);

            foreach (var e in detectEntitiesResponse.Entities)
            {
                Console.WriteLine($"Text: {e.Text}, Type: {e.Type}, Score:
{e.Score}, BeginOffset: {e.BeginOffset}, EndOffset: {e.EndOffset}");
            }

            Console.WriteLine("Done");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectEntitiesà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter les entités nommées dans le texte saisi

L'`detect-entities`exemple suivant analyse le texte saisi et renvoie les entités nommées. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
   account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings, \
   we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000. \
   Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

Sortie :

```
{
    "Entities": [
        {
            "Score": 0.9994556307792664,
            "Type": "PERSON",
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9981022477149963,
            "Type": "PERSON",
            "Text": "John",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.9986887574195862,
            "Type": "ORGANIZATION",
            "Text": "AnyCompany Financial Services, LLC",
            "BeginOffset": 33,
            "EndOffset": 67
        },
        {
            "Score": 0.9959119558334351,
            "Type": "OTHER",
            "Text": "1111-XXXX-1111-XXXX",
```

```
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9708039164543152,
            "Type": "QUANTITY",
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9987268447875977,
            "Type": "DATE",
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9858865737915039,
            "Type": "OTHER",
            "Text": "XXXXXX1111",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9700471758842468,
            "Type": "OTHER",
            "Text": "XXXXX0000",
            "BeginOffset": 306,
            "EndOffset": 315
        },
        {
            "Score": 0.9591118693351746,
            "Type": "ORGANIZATION",
            "Text": "Sunshine Spa",
            "BeginOffset": 340,
            "EndOffset": 352
        },
        {
            "Score": 0.9797496795654297,
            "Type": "LOCATION",
            "Text": "123 Main St",
            "BeginOffset": 354,
            "EndOffset": 365
```

```
        },
        {
            "Score": 0.994929313659668,
            "Type": "PERSON",
            "Text": "Alice",
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9949769377708435,
            "Type": "OTHER",
            "Text": "AnySpa@example.com",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
}
```

Pour plus d'informations, consultez Entities dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DetectEntitiesla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.DetectEntitiesRequest;
import software.amazon.awssdk.services.comprehend.model.DetectEntitiesResponse;
import software.amazon.awssdk.services.comprehend.model.Entity;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import java.util.List;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetectEntities {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
 July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
 blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
 Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
                .region(region)
                .build();

        System.out.println("Calling DetectEntities");
        detectAllEntities(comClient, text);
        comClient.close();
    }

    public static void detectAllEntities(ComprehendClient comClient, String text)
 {
        try {
            DetectEntitiesRequest detectEntitiesRequest =
 DetectEntitiesRequest.builder()
                    .text(text)
                    .languageCode("en")
                    .build();

            DetectEntitiesResponse detectEntitiesResult =
 comClient.detectEntities(detectEntitiesRequest);
            List<Entity> entList = detectEntitiesResult.entities();
            for (Entity entity : entList) {
                System.out.println("Entity text is " + entity.text());
            }

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectEntitiesà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_entities(self, text, language_code):
        """
        Detects entities in a document. Entities can be things like people and
 places
        or other common terms.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of entities along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_entities(
                Text=text, LanguageCode=language_code
```

```
            )
            entities = response["Entities"]
            logger.info("Detected %s entities.", len(entities))
        except ClientError:
            logger.exception("Couldn't detect entities.")
            raise
        else:
            return entities
```

- Pour plus de détails sur l'API, consultez DetectEntitiesle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectKeyPhrases`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectKeyPhrases`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
```

```
    using Amazon.Comprehend;
    using Amazon.Comprehend.Model;

    /// <summary>
    /// This example shows how to use the Amazon Comprehend service to
    /// search text for key phrases.
    /// </summary>
    public static class DetectKeyPhrase
    {
        /// <summary>
        /// This method calls the Amazon Comprehend method DetectKeyPhrasesAsync
        /// to detect any key phrases in the sample text.
        /// </summary>
        public static async Task Main()
        {
            string text = "It is raining today in Seattle";

            var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

            // Call DetectKeyPhrases API
            Console.WriteLine("Calling DetectKeyPhrases");
            var detectKeyPhrasesRequest = new DetectKeyPhrasesRequest()
            {
                Text = text,
                LanguageCode = "en",
            };
            var detectKeyPhrasesResponse = await
comprehendClient.DetectKeyPhrasesAsync(detectKeyPhrasesRequest);
            foreach (var kp in detectKeyPhrasesResponse.KeyPhrases)
            {
                Console.WriteLine($"Text: {kp.Text}, Score: {kp.Score},
BeginOffset: {kp.BeginOffset}, EndOffset: {kp.EndOffset}");
            }

            Console.WriteLine("Done");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectKeyPhrasesà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter les phrases clés dans le texte saisi

L'`detect-key-phrases`exemple suivant analyse le texte saisi et identifie les phrases nominales clés. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-key-phrases \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
        account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due
 by July 31st. Based on your autopay settings, \
        we will withdraw your payment on the due date from your bank account
 number XXXXXX1111 with the routing number XXXXX0000. \
        Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments
 to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "KeyPhrases": [
        {
            "Score": 0.8996376395225525,
            "Text": "Zhang Wei",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9992469549179077,
            "Text": "John",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.988385021686554,
            "Text": "Your AnyCompany Financial Services",
            "BeginOffset": 28,
            "EndOffset": 62
        },
```

```
        {
            "Score": 0.8740853071212769,
            "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
            "BeginOffset": 64,
            "EndOffset": 107
        },
        {
            "Score": 0.9999437928199768,
            "Text": "a minimum payment",
            "BeginOffset": 112,
            "EndOffset": 129
        },
        {
            "Score": 0.9998900890350342,
            "Text": ".53",
            "BeginOffset": 133,
            "EndOffset": 136
        },
        {
            "Score": 0.9979453086853027,
            "Text": "July 31st",
            "BeginOffset": 152,
            "EndOffset": 161
        },
        {
            "Score": 0.9983011484146118,
            "Text": "your autopay settings",
            "BeginOffset": 172,
            "EndOffset": 193
        },
        {
            "Score": 0.9996572136878967,
            "Text": "your payment",
            "BeginOffset": 211,
            "EndOffset": 223
        },
        {
            "Score": 0.9995037317276001,
            "Text": "the due date",
            "BeginOffset": 227,
            "EndOffset": 239
        },
        {
            "Score": 0.9702621698379517,
```

```
            "Text": "your bank account number XXXXXX1111",
            "BeginOffset": 245,
            "EndOffset": 280
        },
        {
            "Score": 0.9179925918579102,
            "Text": "the routing number XXXXX0000.Customer feedback",
            "BeginOffset": 286,
            "EndOffset": 332
        },
        {
            "Score": 0.9978160858154297,
            "Text": "Sunshine Spa",
            "BeginOffset": 337,
            "EndOffset": 349
        },
        {
            "Score": 0.9706913232803345,
            "Text": "123 Main St",
            "BeginOffset": 351,
            "EndOffset": 362
        },
        {
            "Score": 0.9941995143890381,
            "Text": "comments",
            "BeginOffset": 379,
            "EndOffset": 387
        },
        {
            "Score": 0.9759287238121033,
            "Text": "Alice",
            "BeginOffset": 391,
            "EndOffset": 396
        },
        {
            "Score": 0.8376792669296265,
            "Text": "AnySpa@example.com",
            "BeginOffset": 400,
            "EndOffset": 415
        }
    ]
}
```

Pour plus d'informations, consultez les phrases clés du manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DetectKeyPhraseslа section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.DetectKeyPhrasesRequest;
import software.amazon.awssdk.services.comprehend.model.DetectKeyPhrasesResponse;
import software.amazon.awssdk.services.comprehend.model.KeyPhrase;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetectKeyPhrases {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
 July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
 blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
 Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
                .region(region)
```

```
                .build();

        System.out.println("Calling DetectKeyPhrases");
        detectAllKeyPhrases(comClient, text);
        comClient.close();
    }

    public static void detectAllKeyPhrases(ComprehendClient comClient, String
 text) {
        try {
            DetectKeyPhrasesRequest detectKeyPhrasesRequest =
 DetectKeyPhrasesRequest.builder()
                    .text(text)
                    .languageCode("en")
                    .build();

            DetectKeyPhrasesResponse detectKeyPhrasesResult =
 comClient.detectKeyPhrases(detectKeyPhrasesRequest);
            List<KeyPhrase> phraseList = detectKeyPhrasesResult.keyPhrases();
            for (KeyPhrase keyPhrase : phraseList) {
                System.out.println("Key phrase text is " + keyPhrase.text());
            }

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectKeyPhrasesà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_key_phrases(self, text, language_code):
        """
        Detects key phrases in a document. A key phrase is typically a noun and
 its
        modifiers.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of key phrases along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_key_phrases(
                Text=text, LanguageCode=language_code
            )
            phrases = response["KeyPhrases"]
            logger.info("Detected %s phrases.", len(phrases))
        except ClientError:
            logger.exception("Couldn't detect phrases.")
            raise
        else:
            return phrases
```

- Pour plus de détails sur l'API, consultez DetectKeyPhraseslle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectPiiEntities`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectPiiEntities`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use the Amazon Comprehend service to find
/// personally identifiable information (PII) within text submitted to the
/// DetectPiiEntitiesAsync method.
/// </summary>
public class DetectingPII
```

```
    {
        /// <summary>
        /// This method calls the DetectPiiEntitiesAsync method to locate any
        /// personally dientifiable information within the supplied text.
        /// </summary>
        public static async Task Main()
        {
            var comprehendClient = new AmazonComprehendClient();
            var text = @"Hello Paul Santos. The latest statement for your
                         credit card account 1111-0000-1111-0000 was
                         mailed to 123 Any Street, Seattle, WA 98109.";

            var request = new DetectPiiEntitiesRequest
            {
                Text = text,
                LanguageCode = "EN",
            };

            var response = await
comprehendClient.DetectPiiEntitiesAsync(request);

            if (response.Entities.Count > 0)
            {
                foreach (var entity in response.Entities)
                {
                    var entityValue = text.Substring(entity.BeginOffset,
entity.EndOffset - entity.BeginOffset);
                    Console.WriteLine($"{entity.Type}: {entityValue}");
                }
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectPiiEntitiesà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter les entités pii dans le texte saisi

L'`detect-pii-entities`exemple suivant analyse le texte saisi et identifie les entités contenant des informations personnelles identifiables (PII). Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card \
        account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due
 by July 31st. Based on your autopay settings, \
        we will withdraw your payment on the due date from your bank account
 number XXXXXX1111 with the routing number XXXXX0000. \
        Customer feedback for Sunshine Spa, 123 Main St, Anywhere. Send comments
 to Alice at AnySpa@example.com."
```

Sortie :

```
{
    "Entities": [
        {
            "Score": 0.9998322129249573,
            "Type": "NAME",
            "BeginOffset": 6,
            "EndOffset": 15
        },
        {
            "Score": 0.9998878240585327,
            "Type": "NAME",
            "BeginOffset": 22,
            "EndOffset": 26
        },
        {
            "Score": 0.9994089603424072,
            "Type": "CREDIT_DEBIT_NUMBER",
            "BeginOffset": 88,
            "EndOffset": 107
        },
        {
            "Score": 0.9999760985374451,
            "Type": "DATE_TIME",
            "BeginOffset": 152,
            "EndOffset": 161
        },
```

```
        {
            "Score": 0.9999449253082275,
            "Type": "BANK_ACCOUNT_NUMBER",
            "BeginOffset": 271,
            "EndOffset": 281
        },
        {
            "Score": 0.9999847412109375,
            "Type": "BANK_ROUTING",
            "BeginOffset": 306,
            "EndOffset": 315
        },
        {
            "Score": 0.999925434589386,
            "Type": "ADDRESS",
            "BeginOffset": 354,
            "EndOffset": 365
        },
        {
            "Score": 0.9989161491394043,
            "Type": "NAME",
            "BeginOffset": 394,
            "EndOffset": 399
        },
        {
            "Score": 0.9994171857833862,
            "Type": "EMAIL",
            "BeginOffset": 403,
            "EndOffset": 418
        }
    ]
}
```

Pour plus d'informations, consultez la section Informations personnelles identifiables (PII) dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DetectPiiEntitiesla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_pii(self, text, language_code):
        """
        Detects personally identifiable information (PII) in a document. PII can
 be
        things like names, account numbers, or addresses.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of PII entities along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_pii_entities(
                Text=text, LanguageCode=language_code
            )
            entities = response["Entities"]
            logger.info("Detected %s PII entities.", len(entities))
        except ClientError:
            logger.exception("Couldn't detect PII entities.")
            raise
        else:
            return entities
```

- Pour plus de détails sur l'API, consultez DetectPiiEntitiesle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectSentiment`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectSentiment`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to detect the overall sentiment of the supplied
/// text using the Amazon Comprehend service.
/// </summary>
public static class DetectSentiment
{
```

```
        /// <summary>
        /// This method calls the DetetectSentimentAsync method to analyze the
        /// supplied text and determine the overal sentiment.
        /// </summary>
        public static async Task Main()
        {
            string text = "It is raining today in Seattle";

            var comprehendClient = new
AmazonComprehendClient(Amazon.RegionEndpoint.USWest2);

            // Call DetectKeyPhrases API
            Console.WriteLine("Calling DetectSentiment");
            var detectSentimentRequest = new DetectSentimentRequest()
            {
                Text = text,
                LanguageCode = "en",
            };
            var detectSentimentResponse = await
comprehendClient.DetectSentimentAsync(detectSentimentRequest);
            Console.WriteLine($"Sentiment: {detectSentimentResponse.Sentiment}");
            Console.WriteLine("Done");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectSentimentà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter le sentiment d'un texte saisi

L'`detect-sentiment`exemple suivant analyse le texte saisi et renvoie une inférence du sentiment dominant (`POSITIVE`, `NEUTRALMIXED`, ou`NEGATIVE`).

```
aws comprehend detect-sentiment \
    --language-code en \
    --text "It is a beautiful day in Seattle"
```

Sortie :

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.9976957440376282,
        "Negative": 9.653854067437351e-05,
        "Neutral": 0.002169104292988777,
        "Mixed": 3.857641786453314e-05
    }
}
```

Pour plus d'informations, consultez Sentiment dans le guide du développeur Amazon Comprehend

- Pour plus de détails sur l'API, voir DetectSentimentla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import software.amazon.awssdk.services.comprehend.model.DetectSentimentRequest;
import software.amazon.awssdk.services.comprehend.model.DetectSentimentResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetectSentiment {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
 July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
 blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
 Seattle - based companies are Starbucks and Boeing.";
        Region region = Region.US_EAST_1;
        ComprehendClient comClient = ComprehendClient.builder()
                .region(region)
                .build();

        System.out.println("Calling DetectSentiment");
        detectSentiments(comClient, text);
        comClient.close();
    }

    public static void detectSentiments(ComprehendClient comClient, String text)
 {
        try {
            DetectSentimentRequest detectSentimentRequest =
 DetectSentimentRequest.builder()
                    .text(text)
                    .languageCode("en")
                    .build();

            DetectSentimentResponse detectSentimentResult =
 comClient.detectSentiment(detectSentimentRequest);
            System.out.println("The Neutral value is " +
 detectSentimentResult.sentimentScore().neutral());

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectSentimentà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_sentiment(self, text, language_code):
        """
        Detects the overall sentiment expressed in a document. Sentiment can
        be positive, negative, neutral, or a mixture.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The sentiments along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_sentiment(
                Text=text, LanguageCode=language_code
            )
            logger.info("Detected primary sentiment %s.", response["Sentiment"])
        except ClientError:
            logger.exception("Couldn't detect sentiment.")
            raise
        else:
            return response
```

- Pour plus de détails sur l'API, consultez DetectSentimentle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DetectSyntax`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DetectSyntax`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Détecter les éléments du document

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.Comprehend;
using Amazon.Comprehend.Model;

/// <summary>
/// This example shows how to use Amazon Comprehend to detect syntax
/// elements by calling the DetectSyntaxAsync method.
/// </summary>
public class DetectingSyntax
{
    /// <summary>
```

```
        /// This method calls DetectSynaxAsync to identify the syntax elements
        /// in the sample text.
        /// </summary>
        public static async Task Main()
        {
            string text = "It is raining today in Seattle";

            var comprehendClient = new AmazonComprehendClient();

            // Call DetectSyntax API
            Console.WriteLine("Calling DetectSyntaxAsync\n");
            var detectSyntaxRequest = new DetectSyntaxRequest()
            {
                Text = text,
                LanguageCode = "en",
            };
            DetectSyntaxResponse detectSyntaxResponse = await
comprehendClient.DetectSyntaxAsync(detectSyntaxRequest);
            foreach (SyntaxToken s in detectSyntaxResponse.SyntaxTokens)
            {
                Console.WriteLine($"Text: {s.Text}, PartOfSpeech:
{s.PartOfSpeech.Tag}, BeginOffset: {s.BeginOffset}, EndOffset: {s.EndOffset}");
            }

            Console.WriteLine("Done");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DetectSyntaxà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour détecter les parties du discours dans un texte saisi

L'`detect-syntax`exemple suivant analyse la syntaxe du texte saisi et renvoie les différentes parties du discours. Le score de confiance du modèle préentraîné est également généré pour chaque prédiction.

```
aws comprehend detect-syntax \
    --language-code en \
    --text "It is a beautiful day in Seattle."
```

Sortie :

```
{
    "SyntaxTokens": [
        {
            "TokenId": 1,
            "Text": "It",
            "BeginOffset": 0,
            "EndOffset": 2,
            "PartOfSpeech": {
                "Tag": "PRON",
                "Score": 0.9999740719795227
            }
        },
        {
            "TokenId": 2,
            "Text": "is",
            "BeginOffset": 3,
            "EndOffset": 5,
            "PartOfSpeech": {
                "Tag": "VERB",
                "Score": 0.999901294708252
            }
        },
        {
            "TokenId": 3,
            "Text": "a",
            "BeginOffset": 6,
            "EndOffset": 7,
            "PartOfSpeech": {
                "Tag": "DET",
                "Score": 0.9999938607215881
            }
        },
        {
            "TokenId": 4,
            "Text": "beautiful",
            "BeginOffset": 8,
            "EndOffset": 17,
```

```
            "PartOfSpeech": {
                "Tag": "ADJ",
                "Score": 0.9987351894378662
            }
        },
        {
            "TokenId": 5,
            "Text": "day",
            "BeginOffset": 18,
            "EndOffset": 21,
            "PartOfSpeech": {
                "Tag": "NOUN",
                "Score": 0.9999796748161316
            }
        },
        {
            "TokenId": 6,
            "Text": "in",
            "BeginOffset": 22,
            "EndOffset": 24,
            "PartOfSpeech": {
                "Tag": "ADP",
                "Score": 0.9998047947883606
            }
        },
        {
            "TokenId": 7,
            "Text": "Seattle",
            "BeginOffset": 25,
            "EndOffset": 32,
            "PartOfSpeech": {
                "Tag": "PROPN",
                "Score": 0.9940530061721802
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse syntaxique dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir DetectSyntaxla section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.comprehend.ComprehendClient;
import software.amazon.awssdk.services.comprehend.model.ComprehendException;
import software.amazon.awssdk.services.comprehend.model.DetectSyntaxRequest;
import software.amazon.awssdk.services.comprehend.model.DetectSyntaxResponse;
import software.amazon.awssdk.services.comprehend.model.SyntaxToken;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DetectSyntax {
    public static void main(String[] args) {
        String text = "Amazon.com, Inc. is located in Seattle, WA and was founded
 July 5th, 1994 by Jeff Bezos, allowing customers to buy everything from books to
 blenders. Seattle is north of Portland and south of Vancouver, BC. Other notable
 Seattle - based companies are Starbucks and Boeing.";
       Region region = Region.US_EAST_1;
       ComprehendClient comClient = ComprehendClient.builder()
               .region(region)
               .build();

       System.out.println("Calling DetectSyntax");
       detectAllSyntax(comClient, text);
       comClient.close();
    }
```

```
    public static void detectAllSyntax(ComprehendClient comClient, String text) {
        try {
            DetectSyntaxRequest detectSyntaxRequest =
 DetectSyntaxRequest.builder()
                    .text(text)
                    .languageCode("en")
                    .build();

            DetectSyntaxResponse detectSyntaxResult =
 comClient.detectSyntax(detectSyntaxRequest);
            List<SyntaxToken> syntaxTokens = detectSyntaxResult.syntaxTokens();
            for (SyntaxToken token : syntaxTokens) {
                System.out.println("Language is " + token.text());
                System.out.println("Part of speech is " +
 token.partOfSpeech().tagAsString());
            }

        } catch (ComprehendException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DetectSyntaxà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""
```

```
    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_syntax(self, text, language_code):
        """
        Detects syntactical elements of a document. Syntax tokens are portions of
        text along with their use as parts of speech, such as nouns, verbs, and
        interjections.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of syntax tokens along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_syntax(
                Text=text, LanguageCode=language_code
            )
            tokens = response["SyntaxTokens"]
            logger.info("Detected %s syntax tokens.", len(tokens))
        except ClientError:
            logger.exception("Couldn't detect syntax.")
            raise
        else:
            return tokens
```

- Pour plus de détails sur l'API, consultez DetectSyntaxle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListDocumentClassificationJobs`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListDocumentClassificationJobs`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour répertorier tous les travaux de classification de documents

L'`list-document-classification-jobs`exemple suivant répertorie toutes les tâches de classification de documents.

```
aws comprehend list-document-classification-jobs
```

Sortie :

```
{
    "DocumentClassificationJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a11EXAMPLE",
            "JobName": "exampleclassificationjob",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-14T17:09:51.788000+00:00",
            "EndTime": "2023-06-14T17:15:58.582000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
```

```
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-e758dd56b824aa717ceab551f11749fb/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "exampleclassificationjob2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-14T17:22:39.829000+00:00",
            "EndTime": "2023-06-14T17:28:46.107000+00:00",
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:1234567890101:document-classifier/mymodel/version/12",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket/jobdata/",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/1234567890101-CLN-123456abcdeb0e11022f22a1EXAMPLE2/output/
output.tar.gz"
            },
            "DataAccessRoleArn": "arn:aws:iam::1234567890101:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir ListDocumentClassificationJobsla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def list_jobs(self):
        """
        Lists the classification jobs for the current account.

        :return: The list of jobs.
        """
        try:
            response = self.comprehend_client.list_document_classification_jobs()
            jobs = response["DocumentClassificationJobPropertiesList"]
            logger.info("Got %s document classification jobs.", len(jobs))
        except ClientError:
            logger.exception(
                "Couldn't get document classification jobs.",
            )
            raise
        else:
            return jobs
```

- Pour plus de détails sur l'API, consultez ListDocumentClassificationJobsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListDocumentClassifiers`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListDocumentClassifiers`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour afficher la liste de tous les classificateurs de documents

L'`list-document-classifiers`exemple suivant répertorie tous les modèles de classificateur de documents entraînés et en cours de formation.

```
aws comprehend list-document-classifiers
```

Sortie :

```
{
    "DocumentClassifierPropertiesList": [
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier1",
            "LanguageCode": "en",
            "Status": "TRAINED",
            "SubmitTime": "2023-06-13T19:04:15.735000+00:00",
            "EndTime": "2023-06-13T19:42:31.752000+00:00",
            "TrainingStartTime": "2023-06-13T19:08:20.114000+00:00",
            "TrainingEndTime": "2023-06-13T19:41:35.080000+00:00",
```

```
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "ClassifierMetadata": {
                "NumberOfLabels": 3,
                "NumberOfTrainedDocuments": 5016,
                "NumberOfTestDocuments": 557,
                "EvaluationMetrics": {
                    "Accuracy": 0.9856,
                    "Precision": 0.9919,
                    "Recall": 0.9459,
                    "F1Score": 0.9673,
                    "MicroPrecision": 0.9856,
                    "MicroRecall": 0.9856,
                    "MicroF1Score": 0.9856,
                    "HammingLoss": 0.0144
                }
            },
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        },
        {
            "DocumentClassifierArn": "arn:aws:comprehend:us-
west-2:111122223333:document-classifier/exampleclassifier2",
            "LanguageCode": "en",
            "Status": "TRAINING",
            "SubmitTime": "2023-06-13T21:20:28.690000+00:00",
            "InputDataConfig": {
                "DataFormat": "COMPREHEND_CSV",
                "S3Uri": "s3://amzn-s3-demo-bucket/trainingdata"
            },
            "OutputDataConfig": {},
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-testorle",
            "Mode": "MULTI_CLASS"
        }
    ]
}
```

Pour plus d'informations, consultez la section Création et gestion de modèles personnalisés dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir ListDocumentClassifiersla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def list(self):
        """
        Lists custom classifiers for the current account.

        :return: The list of classifiers.
        """
        try:
            response = self.comprehend_client.list_document_classifiers()
            classifiers = response["DocumentClassifierPropertiesList"]
            logger.info("Got %s classifiers.", len(classifiers))
        except ClientError:
            logger.exception(
                "Couldn't get classifiers.",
            )
            raise
```

```
        else:
            return classifiers
```

- Pour plus de détails sur l'API, consultez ListDocumentClassifiersle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListTopicsDetectionJobs`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListTopicsDetectionJobs`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Exécuter une tâche de modélisation de rubrique sur des exemples de données

CLI

AWS CLI

Pour répertorier toutes les tâches de détection de sujets

L'`list-topics-detection-jobs`exemple suivant répertorie toutes les tâches de détection de sujets asynchrones en cours et terminées.

```
aws comprehend list-topics-detection-jobs
```

Sortie :

```
{
    "TopicsDetectionJobPropertiesList": [
        {
            "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
```

```
            "JobName" "topic-analysis-1"
            "JobStatus": "IN_PROGRESS",
            "SubmitTime": "2023-06-09T18:40:35.384000+00:00",
            "EndTime": "2023-06-09T18:46:41.936000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a11EXAMPLE/output/
output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "topic-analysis-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T18:44:43.414000+00:00",
            "EndTime": "2023-06-09T18:50:50.872000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE2/output/
output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE3",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a1EXAMPLE3",
            "JobName": "topic-analysis-2",
            "JobStatus": "IN_PROGRESS",
```

```
            "SubmitTime": "2023-06-09T18:50:56.737000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-bucket",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://amzn-s3-demo-destination-bucket/
thefolder/111122223333-TOPICS-123456abcdeb0e11022f22a1EXAMPLE3/output/
output.tar.gz"
            },
            "NumberOfTopics": 10,
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

Pour plus d'informations, consultez la section Analyse asynchrone pour Amazon Comprehend Insights dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir ListTopicsDetectionJobsla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
```

```
    def list_jobs(self):
        """
        Lists topic modeling jobs for the current account.

        :return: The list of jobs.
        """
        try:
            response = self.comprehend_client.list_topics_detection_jobs()
            jobs = response["TopicsDetectionJobPropertiesList"]
            logger.info("Got %s topic detection jobs.", len(jobs))
        except ClientError:
            logger.exception("Couldn't get topic detection jobs.")
            raise
        else:
            return jobs
```

- Pour plus de détails sur l'API, consultez ListTopicsDetectionJobsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`StartDocumentClassificationJob`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `StartDocumentClassificationJob`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Formez un classificateur personnalisé et classez des documents

CLI

AWS CLI

Pour démarrer une tâche de classification de documents

L'`start-document-classification-job`exemple suivant démarre une tâche de classification de documents avec un modèle personnalisé sur tous les fichiers à l'adresse spécifiée par la `--input-data-config` balise. Dans cet exemple, le compartiment S3 d'entrée contient `SampleSMStext1.txtSampleSMStext2.txt`, et`SampleSMStext3.txt`. Le modèle avait déjà été formé à la classification des documents contenant du spam et des messages non indésirables, ou « spam », des messages SMS. Lorsque le travail est terminé, `output.tar.gz` il est placé à l'emplacement spécifié par la `--output-data-config` balise. `output.tar.gzpredictions.jsonl`contient la liste de classification de chaque document. La sortie Json est imprimée sur une ligne par fichier, mais elle est formatée ici pour des raisons de lisibilité.

```
aws comprehend start-document-classification-job \
    --job-name exampleclassificationjob \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket-INPUT/jobdata/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/"
 \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/mymodel/version/12
```

Contenu de `SampleSMStext1.txt` :

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Contenu de `SampleSMStext2.txt` :

```
"Hi, when do you want me to pick you up from practice?"
```

Contenu de `SampleSMStext3.txt` :

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Sortie :

```
{
    "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Contenu de `predictions.jsonl` :

```
{"File": "SampleSMSText1.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
{"File": "SampleSMStext2.txt", "Line": "0", "Classes": [{"Name": "ham", "Score":
 0.9994}, {"Name": "spam", "Score": 0.0006}]}
{"File": "SampleSMSText3.txt", "Line": "0", "Classes": [{"Name": "spam", "Score":
 0.9999}, {"Name": "ham", "Score": 0.0001}]}
```

Pour plus d'informations, consultez la section Classification personnalisée dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir StartDocumentClassificationJobla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l’exemple complet et découvrez comment le configurer et l’exécuter dans le référentiel d’exemples de code AWS.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None
```

```
    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a classification job. The classifier must be trained or the job
        will fail. Input is read from the specified Amazon S3 input bucket and
        written to the specified output bucket. Output data is stored in a tar
        archive compressed in gzip format. The job runs asynchronously, so you
can
        call `describe_document_classification_job` to get job status until it
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: The Amazon S3 bucket that contains input data.
        :param input_key: The prefix used to find input data in the input
                          bucket. If multiple objects have the same prefix, all
                          of them are used.
        :param input_format: The format of the input data, either one document
per
                             file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                                     grants Comprehend permission to read from
the
                                     input bucket and write to the output bucket.
        :return: Information about the job, including the job ID.
        """
        try:
            response = self.comprehend_client.start_document_classification_job(
                DocumentClassifierArn=self.classifier_arn,
                JobName=job_name,
                InputDataConfig={
                    "S3Uri": f"s3://{input_bucket}/{input_key}",
```

```
                    "InputFormat": input_format.value,
                },
                OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
                DataAccessRoleArn=data_access_role_arn,
            )
            logger.info(
                "Document classification job %s is %s.", job_name,
response["JobStatus"]
            )
        except ClientError:
            logger.exception("Couldn't start classification job %s.", job_name)
            raise
        else:
            return response
```

- Pour plus de détails sur l'API, consultez StartDocumentClassificationJoble AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`StartTopicsDetectionJob`** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `StartTopicsDetectionJob`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Exécuter une tâche de modélisation de rubrique sur des exemples de données

.NET

SDK pour .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Comprehend;
    using Amazon.Comprehend.Model;

    /// <summary>
    /// This example scans the documents in an Amazon Simple Storage Service
    /// (Amazon S3) bucket and analyzes it for topics. The results are stored
    /// in another bucket and then the resulting job properties are displayed
    /// on the screen. This example was created using the AWS SDK for .NEt
    /// version 3.7 and .NET Core version 5.0.
    /// </summary>
    public static class TopicModeling
    {
        /// <summary>
        /// This methos calls a topic detection job by calling the Amazon
        /// Comprehend StartTopicsDetectionJobRequest.
        /// </summary>
        public static async Task Main()
        {
            var comprehendClient = new AmazonComprehendClient();

            string inputS3Uri = "s3://input bucket/input path";
            InputFormat inputDocFormat = InputFormat.ONE_DOC_PER_FILE;
            string outputS3Uri = "s3://output bucket/output path";
            string dataAccessRoleArn = "arn:aws:iam::account ID:role/data access
role";
            int numberOfTopics = 10;

            var startTopicsDetectionJobRequest = new
StartTopicsDetectionJobRequest()
            {
```

```
                InputDataConfig = new InputDataConfig()
                {
                    S3Uri = inputS3Uri,
                    InputFormat = inputDocFormat,
                },
                OutputDataConfig = new OutputDataConfig()
                {
                    S3Uri = outputS3Uri,
                },
                DataAccessRoleArn = dataAccessRoleArn,
                NumberOfTopics = numberOfTopics,
            };

            var startTopicsDetectionJobResponse = await
comprehendClient.StartTopicsDetectionJobAsync(startTopicsDetectionJobRequest);

            var jobId = startTopicsDetectionJobResponse.JobId;
            Console.WriteLine("JobId: " + jobId);

            var describeTopicsDetectionJobRequest = new
DescribeTopicsDetectionJobRequest()
            {
                JobId = jobId,
            };

            var describeTopicsDetectionJobResponse = await
comprehendClient.DescribeTopicsDetectionJobAsync(describeTopicsDetectionJobRequest);

PrintJobProperties(describeTopicsDetectionJobResponse.TopicsDetectionJobProperties);

            var listTopicsDetectionJobsResponse = await
comprehendClient.ListTopicsDetectionJobsAsync(new
ListTopicsDetectionJobsRequest());
            foreach (var props in
listTopicsDetectionJobsResponse.TopicsDetectionJobPropertiesList)
            {
                PrintJobProperties(props);
            }
        }

        /// <summary>
        /// This method is a helper method that displays the job properties
        /// from the call to StartTopicsDetectionJobRequest.
        /// </summary>
```

```
        /// <param name="props">A list of properties from the call to
        /// StartTopicsDetectionJobRequest.</param>
        private static void PrintJobProperties(TopicsDetectionJobProperties
 props)
        {
            Console.WriteLine($"JobId: {props.JobId}, JobName: {props.JobName},
 JobStatus: {props.JobStatus}");
            Console.WriteLine($"NumberOfTopics:
 {props.NumberOfTopics}\nInputS3Uri: {props.InputDataConfig.S3Uri}");
            Console.WriteLine($"InputFormat: {props.InputDataConfig.InputFormat},
 OutputS3Uri: {props.OutputDataConfig.S3Uri}");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous StartTopicsDetectionJobà la section Référence des AWS SDK pour .NET API.

CLI

AWS CLI

Pour démarrer une tâche d'analyse de détection de sujets

L'`start-topics-detection-job`exemple suivant lance une tâche de détection de sujets asynchrones pour tous les fichiers situés à l'adresse spécifiée par la `--input-data-config` balise. Lorsque le travail est terminé, le dossier est placé à l'emplacement spécifié par la `--ouput-data-config` balise. `output` `output`contient topic-terms.csv et doc-topics.csv. Le premier fichier de sortie, topic-terms.csv, est une liste des rubriques de la collection. Pour chaque sujet, la liste inclut, par défaut, les principaux termes par sujet en fonction de leur poids. Le second fichier répertorie `doc-topics.csv` les documents associés à un sujet et la proportion du document qui traite du sujet.

```
aws comprehend start-topics-detection-job \
    --job-name example_topics_detection_job \
    --language-code en \
    --input-data-config "S3Uri=s3://amzn-s3-demo-bucket/" \
    --output-data-config "S3Uri=s3://amzn-s3-demo-destination-bucket/testfolder/"
 \
```

```
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role \
    --language-code en
```

Sortie :

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Pour plus d'informations, consultez la section Modélisation des rubriques dans le manuel Amazon Comprehend Developer Guide.

- Pour plus de détails sur l'API, voir StartTopicsDetectionJobla section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def start_job(
        self,
```

```
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a topic modeling job. Input is read from the specified Amazon S3
        input bucket and written to the specified output bucket. Output data is
stored
        in a tar archive compressed in gzip format. The job runs asynchronously,
so you
        can call `describe_topics_detection_job` to get job status until it
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: An Amazon S3 bucket that contains job input.
        :param input_key: The prefix used to find input data in the input
                          bucket. If multiple objects have the same prefix,
all
                          of them are used.
        :param input_format: The format of the input data, either one document
per
                          file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                                     grants Comprehend permission to read from
the
                                     input bucket and write to the output bucket.
        :return: Information about the job, including the job ID.
        """
        try:
            response = self.comprehend_client.start_topics_detection_job(
                JobName=job_name,
                DataAccessRoleArn=data_access_role_arn,
                InputDataConfig={
                    "S3Uri": f"s3://{input_bucket}/{input_key}",
                    "InputFormat": input_format.value,
                },
                OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
```

```
            )
            logger.info("Started topic modeling job %s.", response["JobId"])
        except ClientError:
            logger.exception("Couldn't start topic modeling job.")
            raise
        else:
            return response
```

- Pour plus de détails sur l'API, consultez StartTopicsDetectionJoble AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios d'utilisation d'Amazon Comprehend AWS SDKs

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans Amazon Comprehend avec. AWS SDKs Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'Amazon Comprehend ou en les combinant à d'autres. Services AWS Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la configuration et l'exécution du code.

Les scénarios ciblent un niveau d'expérience intermédiaire pour vous aider à comprendre les actions de service dans leur contexte.

Exemples

- Créer une application de streaming Amazon Transcribe
- Créez un chatbot Amazon Lex pour engager les visiteurs de votre site Web
- Créez une application Web qui envoie et récupère des messages à l'aide d'Amazon SQS
- Créez une application qui analyse les commentaires des clients et synthétise le son
- Détectez les éléments d'un document avec Amazon Comprehend et un SDK AWS
- Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK
- Exécutez une tâche de modélisation de sujets Amazon Comprehend sur des exemples de données à l'aide d'un SDK AWS

- Formez un classificateur Amazon Comprehend personnalisé et classez des documents à l'aide d'un SDK AWS

Créer une application de streaming Amazon Transcribe

L'exemple de code suivant montre comment créer une application qui enregistre, transcrit et traduit de l'audio en direct en temps réel, et envoie les résultats par e-mail.

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser Amazon Transcribe afin de créer une application qui enregistre, transcrit et traduit de l'audio en direct en temps réel, et envoie les résultats par e-mail à l'aide d'Amazon Simple Email Service (Amazon SES).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Créez un chatbot Amazon Lex pour engager les visiteurs de votre site Web

Les exemples de code suivants montrent comment créer un chatbot pour engager les visiteurs de votre site Web.

Java

SDK pour Java 2.x

Montre comment utiliser l'API Amazon Lex pour créer un Chatbot au sein d'une application Web afin d'engager les visiteurs de votre site Web.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser l'API Amazon Lex pour créer un Chatbot au sein d'une application Web afin d'engager les visiteurs de votre site Web.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet de création d'un chatbot Amazon Lex dans le guide du AWS SDK pour JavaScript développeur.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Créez une application Web qui envoie et récupère des messages à l'aide d'Amazon SQS

Les exemples de code suivants montrent comment créer une application de messagerie à l'aide d'Amazon SQS.

Java

SDK pour Java 2.x

Montre comment utiliser l'API Amazon SQS pour développer une API Spring REST qui envoie et récupère des messages.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon SQS

Kotlin

SDK pour Kotlin

Montre comment utiliser l'API Amazon SQS pour développer une API Spring REST qui envoie et récupère des messages.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon SQS

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Créez une application qui analyse les commentaires des clients et synthétise le son

Les exemples de code suivants montrent comment créer une application qui analyse les cartes de commentaires des clients, les traduit depuis leur langue d'origine, détermine leur sentiment et génère un fichier audio à partir du texte traduit.

.NET

SDK pour .NET

Cet exemple d'application analyse et stocke les cartes de commentaires des clients. Plus précisément, elle répond aux besoins d'un hôtel fictif situé à New York. L'hôtel reçoit les commentaires des clients dans différentes langues sous la forme de cartes de commentaires physiques. Ces commentaires sont chargés dans l'application via un client Web. Après avoir chargé l'image d'une carte de commentaires, les étapes suivantes se déroulent :

- Le texte est extrait de l'image à l'aide d'Amazon Textract.
- Amazon Comprehend détermine le sentiment du texte extrait et sa langue.
- Le texte extrait est traduit en anglais à l'aide d'Amazon Translate.
- Amazon Polly synthétise un fichier audio à partir du texte extrait.

L'application complète peut être déployée avec AWS CDK. Pour le code source et les instructions de déploiement, consultez le projet dans GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK pour Java 2.x

Cet exemple d'application analyse et stocke les cartes de commentaires des clients. Plus précisément, elle répond aux besoins d'un hôtel fictif situé à New York. L'hôtel reçoit les

commentaires des clients dans différentes langues sous la forme de cartes de commentaires physiques. Ces commentaires sont chargés dans l'application via un client Web. Après avoir chargé l'image d'une carte de commentaires, les étapes suivantes se déroulent :

- Le texte est extrait de l'image à l'aide d'Amazon Textract.
- Amazon Comprehend détermine le sentiment du texte extrait et sa langue.
- Le texte extrait est traduit en anglais à l'aide d'Amazon Translate.
- Amazon Polly synthétise un fichier audio à partir du texte extrait.

L'application complète peut être déployée avec AWS CDK. Pour le code source et les instructions de déploiement, consultez le projet dans GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK pour JavaScript (v3)

Cet exemple d'application analyse et stocke les cartes de commentaires des clients. Plus précisément, elle répond aux besoins d'un hôtel fictif situé à New York. L'hôtel reçoit les commentaires des clients dans différentes langues sous la forme de cartes de commentaires physiques. Ces commentaires sont chargés dans l'application via un client Web. Après avoir chargé l'image d'une carte de commentaires, les étapes suivantes se déroulent :

- Le texte est extrait de l'image à l'aide d'Amazon Textract.
- Amazon Comprehend détermine le sentiment du texte extrait et sa langue.
- Le texte extrait est traduit en anglais à l'aide d'Amazon Translate.
- Amazon Polly synthétise un fichier audio à partir du texte extrait.

L'application complète peut être déployée avec AWS CDK. Pour le code source et les instructions de déploiement, consultez le projet dans GitHub. Les extraits suivants montrent comment le AWS SDK pour JavaScript est utilisé dans les fonctions Lambda.

```
import {
  ComprehendClient,
  DetectDominantLanguageCommand,
  DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
 * Determine the language and sentiment of the extracted text.
 *
 * @param {{ source_text: string}} extractTextOutput
 */
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });

  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};
```

```
import {
  DetectDocumentTextCommand,
  TextractClient,
} from "@aws-sdk/client-textract";
```

```
/**
 * Fetch the S3 object from the event and analyze it using Amazon Textract.
 *
 * @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
 eventBridgeS3Event
 */
export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/
textract/latest/dg/API_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );

  return extractedWords.join(" ");
};
```

```
import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string}}
 sourceDestinationConfig
 */
```

```
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};
```

```
import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string}}
 textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});
```

```
  const translateCommand = new TranslateTextCommand({
    SourceLanguageCode: textAndSourceLanguage.source_language_code,
    TargetLanguageCode: "en",
    Text: textAndSourceLanguage.extracted_text,
  });

  const { TranslatedText } = await translateClient.send(translateCommand);

  return { translated_text: TranslatedText };
};
```

Les services utilisés dans cet exemple

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

Kit SDK pour Ruby

Cet exemple d'application analyse et stocke les cartes de commentaires des clients. Plus précisément, elle répond aux besoins d'un hôtel fictif situé à New York. L'hôtel reçoit les commentaires des clients dans différentes langues sous la forme de cartes de commentaires physiques. Ces commentaires sont chargés dans l'application via un client Web. Après avoir chargé l'image d'une carte de commentaires, les étapes suivantes se déroulent :

- Le texte est extrait de l'image à l'aide d'Amazon Textract.
- Amazon Comprehend détermine le sentiment du texte extrait et sa langue.
- Le texte extrait est traduit en anglais à l'aide d'Amazon Translate.
- Amazon Polly synthétise un fichier audio à partir du texte extrait.

L'application complète peut être déployée avec AWS CDK. Pour le code source et les instructions de déploiement, consultez le projet dans GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend

- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détectez les éléments d'un document avec Amazon Comprehend et un SDK AWS

L'exemple de code suivant illustre comment :

- Détectez les langues, les entités et les phrases clés d'un document.
- Détectez les informations personnelles identifiables (PII) dans un document.
- Détectez le sentiment d'un document.
- Détecte les éléments de syntaxe dans un document.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une classe qui englobe les actions Amazon Comprehend.

```
import logging
from pprint import pprint
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
```

```
class ComprehendDetect:
    """Encapsulates Comprehend detection functions."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def detect_languages(self, text):
        """
        Detects languages used in a document.

        :param text: The document to inspect.
        :return: The list of languages along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_dominant_language(Text=text)
            languages = response["Languages"]
            logger.info("Detected %s languages.", len(languages))
        except ClientError:
            logger.exception("Couldn't detect languages.")
            raise
        else:
            return languages


    def detect_entities(self, text, language_code):
        """
        Detects entities in a document. Entities can be things like people and
 places
        or other common terms.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of entities along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_entities(
                Text=text, LanguageCode=language_code
            )
            entities = response["Entities"]
```

```
            logger.info("Detected %s entities.", len(entities))
        except ClientError:
            logger.exception("Couldn't detect entities.")
            raise
        else:
            return entities


    def detect_key_phrases(self, text, language_code):
        """
        Detects key phrases in a document. A key phrase is typically a noun and
its
        modifiers.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of key phrases along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_key_phrases(
                Text=text, LanguageCode=language_code
            )
            phrases = response["KeyPhrases"]
            logger.info("Detected %s phrases.", len(phrases))
        except ClientError:
            logger.exception("Couldn't detect phrases.")
            raise
        else:
            return phrases


    def detect_pii(self, text, language_code):
        """
        Detects personally identifiable information (PII) in a document. PII can
be
        things like names, account numbers, or addresses.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of PII entities along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_pii_entities(
                Text=text, LanguageCode=language_code
```

```
            )
            entities = response["Entities"]
            logger.info("Detected %s PII entities.", len(entities))
        except ClientError:
            logger.exception("Couldn't detect PII entities.")
            raise
        else:
            return entities


    def detect_sentiment(self, text, language_code):
        """
        Detects the overall sentiment expressed in a document. Sentiment can
        be positive, negative, neutral, or a mixture.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The sentiments along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_sentiment(
                Text=text, LanguageCode=language_code
            )
            logger.info("Detected primary sentiment %s.", response["Sentiment"])
        except ClientError:
            logger.exception("Couldn't detect sentiment.")
            raise
        else:
            return response


    def detect_syntax(self, text, language_code):
        """
        Detects syntactical elements of a document. Syntax tokens are portions of
        text along with their use as parts of speech, such as nouns, verbs, and
        interjections.

        :param text: The document to inspect.
        :param language_code: The language of the document.
        :return: The list of syntax tokens along with their confidence scores.
        """
        try:
            response = self.comprehend_client.detect_syntax(
                Text=text, LanguageCode=language_code
```

```
            )
            tokens = response["SyntaxTokens"]
            logger.info("Detected %s syntax tokens.", len(tokens))
        except ClientError:
            logger.exception("Couldn't detect syntax.")
            raise
        else:
            return tokens
```

Appelez des fonctions de la classe wrapper pour détecter des entités, des phrases, etc. dans un document.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend detection demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    comp_detect = ComprehendDetect(boto3.client("comprehend"))
    with open("detect_sample.txt") as sample_file:
        sample_text = sample_file.read()

    demo_size = 3

    print("Sample text used for this demo:")
    print("-" * 88)
    print(sample_text)
    print("-" * 88)

    print("Detecting languages.")
    languages = comp_detect.detect_languages(sample_text)
    pprint(languages)
    lang_code = languages[0]["LanguageCode"]

    print("Detecting entities.")
    entities = comp_detect.detect_entities(sample_text, lang_code)
    print(f"The first {demo_size} are:")
    pprint(entities[:demo_size])
```

```
    print("Detecting key phrases.")
    phrases = comp_detect.detect_key_phrases(sample_text, lang_code)
    print(f"The first {demo_size} are:")
    pprint(phrases[:demo_size])

    print("Detecting personally identifiable information (PII).")
    pii_entities = comp_detect.detect_pii(sample_text, lang_code)
    print(f"The first {demo_size} are:")
    pprint(pii_entities[:demo_size])

    print("Detecting sentiment.")
    sentiment = comp_detect.detect_sentiment(sample_text, lang_code)
    print(f"Sentiment: {sentiment['Sentiment']}")
    print("SentimentScore:")
    pprint(sentiment["SentimentScore"])

    print("Detecting syntax elements.")
    syntax_tokens = comp_detect.detect_syntax(sample_text, lang_code)
    print(f"The first {demo_size} are:")
    pprint(syntax_tokens[:demo_size])

    print("Thanks for watching!")
    print("-" * 88)
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - DetectDominantLanguage
  - DetectEntities
  - DetectKeyPhrases
  - DetectPiiEntities
  - DetectSentiment
  - DetectSyntax

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK

L'exemple de code suivant montre comment utiliser Amazon Comprehend pour détecter des entités dans du texte extrait par Amazon Textract à partir d'une image stockée dans Amazon S3.

Python

SDK pour Python (Boto3)

Montre comment utiliser le AWS SDK pour Python (Boto3) dans un bloc-notes Jupyter pour détecter des entités dans du texte extrait d'une image. Cet exemple utilise Amazon Textract pour extraire le texte d'une image stockée dans Amazon Simple Storage Service (Amazon S3) et Amazon Comprehend pour détecter les entités dans le texte extrait.

Cet exemple est un carnet Jupyter et doit être exécuté dans un environnement qui peut accueillir des carnets. Pour savoir comment exécuter cet exemple à l'aide d'Amazon SageMaker AI, consultez les instructions du TextractAndComprehendNotebookfichier .ipynb.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exécutez une tâche de modélisation de sujets Amazon Comprehend sur des exemples de données à l'aide d'un SDK AWS

L’exemple de code suivant illustre comment :

- Exécutez une tâche de modélisation de sujets Amazon Comprehend sur des exemples de données.
- Obtenez des informations sur le poste.

- Extrayez les données de sortie des tâches depuis Amazon S3.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une classe wrapper pour appeler les actions de modélisation de sujets Amazon Comprehend.

```
class ComprehendTopicModeler:
    """Encapsulates a Comprehend topic modeler."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client


    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a topic modeling job. Input is read from the specified Amazon S3
        input bucket and written to the specified output bucket. Output data is
 stored
        in a tar archive compressed in gzip format. The job runs asynchronously,
 so you
        can call `describe_topics_detection_job` to get job status until it
```

```
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: An Amazon S3 bucket that contains job input.
        :param input_key: The prefix used to find input data in the input
                          bucket. If multiple objects have the same prefix,
all
                          of them are used.
        :param input_format: The format of the input data, either one document
per
                          file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                                     grants Comprehend permission to read from
the
                                     input bucket and write to the output bucket.
        :return: Information about the job, including the job ID.
        """
        try:
            response = self.comprehend_client.start_topics_detection_job(
                JobName=job_name,
                DataAccessRoleArn=data_access_role_arn,
                InputDataConfig={
                    "S3Uri": f"s3://{input_bucket}/{input_key}",
                    "InputFormat": input_format.value,
                },
                OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
            )
            logger.info("Started topic modeling job %s.", response["JobId"])
        except ClientError:
            logger.exception("Couldn't start topic modeling job.")
            raise
        else:
            return response


    def describe_job(self, job_id):
        """
        Gets metadata about a topic modeling job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
```

```
        """
        try:
            response = self.comprehend_client.describe_topics_detection_job(
                JobId=job_id
            )
            job = response["TopicsDetectionJobProperties"]
            logger.info("Got topic detection job %s.", job_id)
        except ClientError:
            logger.exception("Couldn't get topic detection job %s.", job_id)
            raise
        else:
            return job


    def list_jobs(self):
        """
        Lists topic modeling jobs for the current account.

        :return: The list of jobs.
        """
        try:
            response = self.comprehend_client.list_topics_detection_jobs()
            jobs = response["TopicsDetectionJobPropertiesList"]
            logger.info("Got %s topic detection jobs.", len(jobs))
        except ClientError:
            logger.exception("Couldn't get topic detection jobs.")
            raise
        else:
            return jobs
```

Utilisez la classe wrapper pour exécuter une tâche de modélisation de rubrique et obtenir des données de tâche.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend topic modeling demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
```

```
    input_prefix = "input/"
    output_prefix = "output/"
    demo_resources = ComprehendDemoResources(
        boto3.resource("s3"), boto3.resource("iam")
    )
    topic_modeler = ComprehendTopicModeler(boto3.client("comprehend"))

    print("Setting up storage and security resources needed for the demo.")
    demo_resources.setup("comprehend-topic-modeler-demo")
    print("Copying sample data from public bucket into input bucket.")
    demo_resources.bucket.copy(
        {"Bucket": "public-sample-us-west-2", "Key": "TopicModeling/Sample.txt"},
        f"{input_prefix}sample.txt",
    )

    print("Starting topic modeling job on sample data.")
    job_info = topic_modeler.start_job(
        "demo-topic-modeling-job",
        demo_resources.bucket.name,
        input_prefix,
        JobInputFormat.per_line,
        demo_resources.bucket.name,
        output_prefix,
        demo_resources.data_access_role.arn,
    )

    print(
        f"Waiting for job {job_info['JobId']} to complete. This typically takes "
        f"20 - 30 minutes."
    )
    job_waiter = JobCompleteWaiter(topic_modeler.comprehend_client)
    job_waiter.wait(job_info["JobId"])

    job = topic_modeler.describe_job(job_info["JobId"])
    print(f"Job {job['JobId']} complete:")
    pprint(job)

    print(
        f"Getting job output data from the output Amazon S3 bucket: "
        f"{job['OutputDataConfig']['S3Uri']}."
    )
    job_output = demo_resources.extract_job_output(job)
    lines = 10
    print(f"First {lines} lines of document topics output:")
```

```
    pprint(job_output["doc-topics.csv"]["data"][:lines])
    print(f"First {lines} lines of terms output:")
    pprint(job_output["topic-terms.csv"]["data"][:lines])

    print("Cleaning up resources created for the demo.")
    demo_resources.cleanup()

    print("Thanks for watching!")
    print("-" * 88)
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - DescribeTopicsDetectionJob
  - ListTopicsDetectionJobs
  - StartTopicsDetectionJob

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Formez un classificateur Amazon Comprehend personnalisé et classez des documents à l'aide d'un SDK AWS

L'exemple de code suivant illustre comment :

- Créez un classificateur multi-étiquettes Amazon Comprehend.
- Entraînez le classificateur sur des échantillons de données.
- Exécutez une tâche de classification sur un deuxième ensemble de données.
- Extrayez les données de sortie de la tâche depuis Amazon S3.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une classe wrapper pour appeler les actions du classificateur de documents Amazon Comprehend.

```
class ComprehendClassifier:
    """Encapsulates an Amazon Comprehend custom classifier."""

    def __init__(self, comprehend_client):
        """
        :param comprehend_client: A Boto3 Comprehend client.
        """
        self.comprehend_client = comprehend_client
        self.classifier_arn = None


    def create(
        self,
        name,
        language_code,
        training_bucket,
        training_key,
        data_access_role_arn,
        mode,
    ):
        """
        Creates a custom classifier. After the classifier is created, it
immediately
        starts training on the data found in the specified Amazon S3 bucket.
Training
        can take 30 minutes or longer. The `describe_document_classifier`
function
        can be used to get training status and returns a status of TRAINED when
the
```

```
        classifier is ready to use.

        :param name: The name of the classifier.
        :param language_code: The language the classifier can operate on.
        :param training_bucket: The Amazon S3 bucket that contains the training
 data.
        :param training_key: The prefix used to find training data in the
 training
                             bucket. If multiple objects have the same prefix,
 all
                             of them are used.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
 that
                                     grants Comprehend permission to read from
 the
                                     training bucket.
        :return: The ARN of the newly created classifier.
        """
        try:
            response = self.comprehend_client.create_document_classifier(
                DocumentClassifierName=name,
                LanguageCode=language_code,
                InputDataConfig={"S3Uri": f"s3://{training_bucket}/
{training_key}"},
                DataAccessRoleArn=data_access_role_arn,
                Mode=mode.value,
            )
            self.classifier_arn = response["DocumentClassifierArn"]
            logger.info("Started classifier creation. Arn is: %s.",
 self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't create classifier %s.", name)
            raise
        else:
            return self.classifier_arn


    def describe(self, classifier_arn=None):
        """
        Gets metadata about a custom classifier, including its current status.

        :param classifier_arn: The ARN of the classifier to look up.
        :return: Metadata about the classifier.
        """
```

```
        if classifier_arn is not None:
            self.classifier_arn = classifier_arn
        try:
            response = self.comprehend_client.describe_document_classifier(
                DocumentClassifierArn=self.classifier_arn
            )
            classifier = response["DocumentClassifierProperties"]
            logger.info("Got classifier %s.", self.classifier_arn)
        except ClientError:
            logger.exception("Couldn't get classifier %s.", self.classifier_arn)
            raise
        else:
            return classifier


    def list(self):
        """
        Lists custom classifiers for the current account.

        :return: The list of classifiers.
        """
        try:
            response = self.comprehend_client.list_document_classifiers()
            classifiers = response["DocumentClassifierPropertiesList"]
            logger.info("Got %s classifiers.", len(classifiers))
        except ClientError:
            logger.exception(
                "Couldn't get classifiers.",
            )
            raise
        else:
            return classifiers


    def delete(self):
        """
        Deletes the classifier.
        """
        try:
            self.comprehend_client.delete_document_classifier(
                DocumentClassifierArn=self.classifier_arn
            )
            logger.info("Deleted classifier %s.", self.classifier_arn)
            self.classifier_arn = None
```

```
        except ClientError:
            logger.exception("Couldn't deleted classifier %s.",
self.classifier_arn)
            raise


    def start_job(
        self,
        job_name,
        input_bucket,
        input_key,
        input_format,
        output_bucket,
        output_key,
        data_access_role_arn,
    ):
        """
        Starts a classification job. The classifier must be trained or the job
        will fail. Input is read from the specified Amazon S3 input bucket and
        written to the specified output bucket. Output data is stored in a tar
        archive compressed in gzip format. The job runs asynchronously, so you
can
        call `describe_document_classification_job` to get job status until it
        returns a status of SUCCEEDED.

        :param job_name: The name of the job.
        :param input_bucket: The Amazon S3 bucket that contains input data.
        :param input_key: The prefix used to find input data in the input
                          bucket. If multiple objects have the same prefix, all
                          of them are used.
        :param input_format: The format of the input data, either one document
per
                             file or one document per line.
        :param output_bucket: The Amazon S3 bucket where output data is written.
        :param output_key: The prefix prepended to the output data.
        :param data_access_role_arn: The Amazon Resource Name (ARN) of a role
that
                                     grants Comprehend permission to read from
the
                                     input bucket and write to the output bucket.
        :return: Information about the job, including the job ID.
        """
        try:
            response = self.comprehend_client.start_document_classification_job(
```

```
                DocumentClassifierArn=self.classifier_arn,
                JobName=job_name,
                InputDataConfig={
                    "S3Uri": f"s3://{input_bucket}/{input_key}",
                    "InputFormat": input_format.value,
                },
                OutputDataConfig={"S3Uri": f"s3://{output_bucket}/{output_key}"},
                DataAccessRoleArn=data_access_role_arn,
            )
            logger.info(
                "Document classification job %s is %s.", job_name,
response["JobStatus"]
            )
        except ClientError:
            logger.exception("Couldn't start classification job %s.", job_name)
            raise
        else:
            return response


    def describe_job(self, job_id):
        """
        Gets metadata about a classification job.

        :param job_id: The ID of the job to look up.
        :return: Metadata about the job.
        """
        try:
            response =
self.comprehend_client.describe_document_classification_job(
                JobId=job_id
            )
            job = response["DocumentClassificationJobProperties"]
            logger.info("Got classification job %s.", job["JobName"])
        except ClientError:
            logger.exception("Couldn't get classification job %s.", job_id)
            raise
        else:
            return job


    def list_jobs(self):
        """
        Lists the classification jobs for the current account.
```

```
        :return: The list of jobs.
        """
        try:
            response = self.comprehend_client.list_document_classification_jobs()
            jobs = response["DocumentClassificationJobPropertiesList"]
            logger.info("Got %s document classification jobs.", len(jobs))
        except ClientError:
            logger.exception(
                "Couldn't get document classification jobs.",
            )
            raise
        else:
            return jobs
```

Créez une classe pour aider à exécuter le scénario.

```
class ClassifierDemo:
    """
    Encapsulates functions used to run the demonstration.
    """

    def __init__(self, demo_resources):
        """
        :param demo_resources: A ComprehendDemoResources class that manages
 resources
                               for the demonstration.
        """
        self.demo_resources = demo_resources
        self.training_prefix = "training/"
        self.input_prefix = "input/"
        self.input_format = JobInputFormat.per_line
        self.output_prefix = "output/"

    def setup(self):
        """Creates AWS resources used by the demo."""
        self.demo_resources.setup("comprehend-classifier-demo")

    def cleanup(self):
        """Deletes AWS resources used by the demo."""
```

```
        self.demo_resources.cleanup()

    @staticmethod
    def _sanitize_text(text):
        """Removes characters that cause errors for the document parser."""
        return text.replace("\r", " ").replace("\n", " ").replace(",", ";")

    @staticmethod
    def _get_issues(query, issue_count):
        """
        Gets issues from GitHub using the specified query parameters.

        :param query: The query string used to request issues from the GitHub
 API.
        :param issue_count: The number of issues to retrieve.
        :return: The list of issues retrieved from GitHub.
        """
        issues = []
        logger.info("Requesting issues from %s?%s.", GITHUB_SEARCH_URL, query)
        response = requests.get(f"{GITHUB_SEARCH_URL}?
{query}&per_page={issue_count}")
        if response.status_code == 200:
            issue_page = response.json()["items"]
            logger.info("Got %s issues.", len(issue_page))
            issues = [
                {
                    "title": ClassifierDemo._sanitize_text(issue["title"]),
                    "body": ClassifierDemo._sanitize_text(issue["body"]),
                    "labels": {label["name"] for label in issue["labels"]},
                }
                for issue in issue_page
            ]
        else:
            logger.error(
                "GitHub returned error code %s with message %s.",
                response.status_code,
                response.json(),
            )
        logger.info("Found %s issues.", len(issues))
        return issues

    def get_training_issues(self, training_labels):
        """
        Gets issues used for training the custom classifier. Training issues are
```

```
        closed issues from the Boto3 repo that have known labels. Comprehend
        requires a minimum of ten training issues per label.

        :param training_labels: The issue labels to use for training.
        :return: The set of issues used for training.
        """
        issues = []
        per_label_count = 15
        for label in training_labels:
            issues += self._get_issues(
                f"q=type:issue+repo:boto/boto3+state:closed+label:{label}",
                per_label_count,
            )
            for issue in issues:
                issue["labels"] = issue["labels"].intersection(training_labels)
        return issues

    def get_input_issues(self, training_labels):
        """
        Gets input issues from GitHub. For demonstration purposes, input issues
        are open issues from the Boto3 repo with known labels, though in practice
        any issue could be submitted to the classifier for labeling.

        :param training_labels: The set of labels to query for.
        :return: The set of issues used for input.
        """
        issues = []
        per_label_count = 5
        for label in training_labels:
            issues += self._get_issues(
                f"q=type:issue+repo:boto/boto3+state:open+label:{label}",
                per_label_count,
            )
        return issues

    def upload_issue_data(self, issues, training=False):
        """
        Uploads issue data to an Amazon S3 bucket, either for training or for
input.
        The data is first put into the format expected by Comprehend. For
training,
        the set of pipe-delimited labels is prepended to each document. For
        input, labels are not sent.
```

```
        :param issues: The set of issues to upload to Amazon S3.
        :param training: Indicates whether the issue data is used for training or
                         input.
        """
        try:
            obj_key = (
                self.training_prefix if training else self.input_prefix
            ) + "issues.txt"
            if training:
                issue_strings = [
                    f"{'|'.join(issue['labels'])},{issue['title']}
{issue['body']}"
                    for issue in issues
                ]
            else:
                issue_strings = [
                    f"{issue['title']} {issue['body']}" for issue in issues
                ]
            issue_bytes = BytesIO("\n".join(issue_strings).encode("utf-8"))
            self.demo_resources.bucket.upload_fileobj(issue_bytes, obj_key)
            logger.info(
                "Uploaded data as %s to bucket %s.",
                obj_key,
                self.demo_resources.bucket.name,
            )
        except ClientError:
            logger.exception(
                "Couldn't upload data to bucket %s.",
self.demo_resources.bucket.name
            )
            raise

    def extract_job_output(self, job):
        """Extracts job output from Amazon S3."""
        return self.demo_resources.extract_job_output(job)

    @staticmethod
    def reconcile_job_output(input_issues, output_dict):
        """
        Reconciles job output with the list of input issues. Because the input
issues
        have known labels, these can be compared with the labels added by the
        classifier to judge the accuracy of the output.
```

```
        :param input_issues: The list of issues used as input.
        :param output_dict: The dictionary of data that is output by the
 classifier.
        :return: The list of reconciled input and output data.
        """
        reconciled = []
        for archive in output_dict.values():
            for line in archive["data"]:
                in_line = int(line["Line"])
                in_labels = input_issues[in_line]["labels"]
                out_labels = {
                    label["Name"]
                    for label in line["Labels"]
                    if float(label["Score"]) > 0.3
                }
                reconciled.append(
                    f"{line['File']}, line {in_line} has labels {in_labels}.\n"
                    f"\tClassifier assigned {out_labels}."
                )
        logger.info("Reconciled input and output labels.")
        return reconciled
```

Formez un classificateur à un ensemble de GitHub problèmes associés à des étiquettes connues, puis envoyez un deuxième ensemble de GitHub problèmes au classificateur afin qu'il puisse les étiqueter.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Comprehend custom document classifier demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    comp_demo = ClassifierDemo(
        ComprehendDemoResources(boto3.resource("s3"), boto3.resource("iam"))
    )
    comp_classifier = ComprehendClassifier(boto3.client("comprehend"))
    classifier_trained_waiter = ClassifierTrainedWaiter(
        comp_classifier.comprehend_client
    )
```

```
    training_labels = {"bug", "feature-request", "dynamodb", "s3"}

    print("Setting up storage and security resources needed for the demo.")
    comp_demo.setup()

    print("Getting training data from GitHub and uploading it to Amazon S3.")
    training_issues = comp_demo.get_training_issues(training_labels)
    comp_demo.upload_issue_data(training_issues, True)

    classifier_name = "doc-example-classifier"
    print(f"Creating document classifier {classifier_name}.")
    comp_classifier.create(
        classifier_name,
        "en",
        comp_demo.demo_resources.bucket.name,
        comp_demo.training_prefix,
        comp_demo.demo_resources.data_access_role.arn,
        ClassifierMode.multi_label,
    )
    print(
        f"Waiting until {classifier_name} is trained. This typically takes "
        f"30-40 minutes."
    )
    classifier_trained_waiter.wait(comp_classifier.classifier_arn)

    print(f"Classifier {classifier_name} is trained:")
    pprint(comp_classifier.describe())

    print("Getting input data from GitHub and uploading it to Amazon S3.")
    input_issues = comp_demo.get_input_issues(training_labels)
    comp_demo.upload_issue_data(input_issues)

    print("Starting classification job on input data.")
    job_info = comp_classifier.start_job(
        "issue_classification_job",
        comp_demo.demo_resources.bucket.name,
        comp_demo.input_prefix,
        comp_demo.input_format,
        comp_demo.demo_resources.bucket.name,
        comp_demo.output_prefix,
        comp_demo.demo_resources.data_access_role.arn,
    )
    print(f"Waiting for job {job_info['JobId']} to complete.")
    job_waiter = JobCompleteWaiter(comp_classifier.comprehend_client)
```

```
    job_waiter.wait(job_info["JobId"])

    job = comp_classifier.describe_job(job_info["JobId"])
    print(f"Job {job['JobId']} complete:")
    pprint(job)

    print(
        f"Getting job output data from Amazon S3: "
        f"{job['OutputDataConfig']['S3Uri']}."
    )
    job_output = comp_demo.extract_job_output(job)
    print("Job output:")
    pprint(job_output)

    print("Reconciling job output with labels from GitHub:")
    reconciled_output = comp_demo.reconcile_job_output(input_issues, job_output)
    print(*reconciled_output, sep="\n")

    answer = input(f"Do you want to delete the classifier {classifier_name} (y/
n)? ")
    if answer.lower() == "y":
        print(f"Deleting {classifier_name}.")
        comp_classifier.delete()

    print("Cleaning up resources created for the demo.")
    comp_demo.cleanup()

    print("Thanks for watching!")
    print("-" * 88)
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - CreateDocumentClassifier
  - DeleteDocumentClassifier
  - DescribeDocumentClassificationJob
  - DescribeDocumentClassifier
  - ListDocumentClassificationJobs
  - ListDocumentClassifiers

- StartDocumentClassificationJob

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation d'Amazon Comprehend avec un SDK AWS. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

La sécurité dans Amazon Comprehend

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée) décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon Comprehend, consultez la section AWS Services concernés par programme de conformitéAWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon Comprehend. Les rubriques suivantes expliquent comment configurer Amazon Comprehend pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Amazon Comprehend.

Rubriques

- Protection des données dans Amazon Comprehend
- Identity and Access Management pour Amazon Comprehend
- Journalisation des appels d'API Amazon Comprehend avec AWS CloudTrail
- Validation de conformité pour Amazon Comprehend
- La résilience dans Amazon Comprehend
- Sécurité de l'infrastructure dans Amazon Comprehend

Protection des données dans Amazon Comprehend

Le modèle de responsabilité AWS partagée Le modèle s'applique à la protection des données dans Amazon Comprehend. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Amazon Comprehend ou une autre entreprise

à Services AWS l'aide de la console, de l'API ou. AWS CLI AWS SDKs Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Rubriques

- Chiffrement KMS dans Amazon Comprehend
- Prévention du problème de l'adjoint confus entre services
- Protégez les emplois en utilisant un Amazon Virtual Private Cloud
- Amazon Comprehend et interface avec les points de terminaison VPC ()AWS PrivateLink

Chiffrement KMS dans Amazon Comprehend

Amazon Comprehend travaille avec AWS Key Management Service (AWS KMS) pour fournir un chiffrement amélioré de vos données. Amazon S3 vous permet déjà de chiffrer vos documents d'entrée lors de la création d'une analyse de texte, d'une modélisation thématique ou d'une tâche Amazon Comprehend personnalisée. L'intégration avec vous AWS KMS permet de chiffrer les données du volume de stockage pour les tâches Start* et Create*, et elle chiffre les résultats de sortie des tâches Start* à l'aide de votre propre clé KMS.

Pour le AWS Management Console, Amazon Comprehend chiffre les modèles personnalisés avec sa propre clé KMS. Dans ce cas AWS CLI, Amazon Comprehend peut chiffrer des modèles personnalisés à l'aide de sa propre clé KMS ou d'une clé gérée par le client (CMK) fournie.

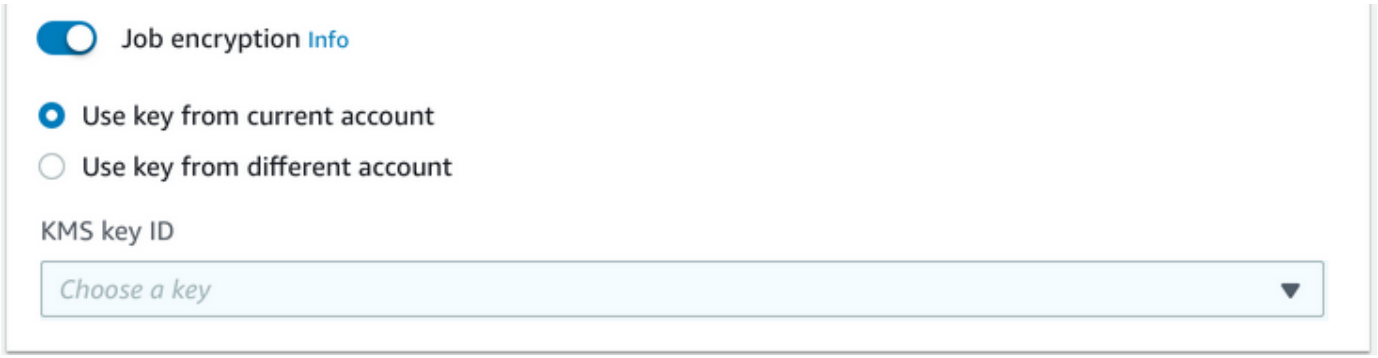
Chiffrement KMS à l'aide du AWS Management Console

Deux options de chiffrement sont disponibles lors de l'utilisation de la console :

- Chiffrement de volume
- Chiffrement des résultats de sortie

Pour activer le chiffrement des volumes

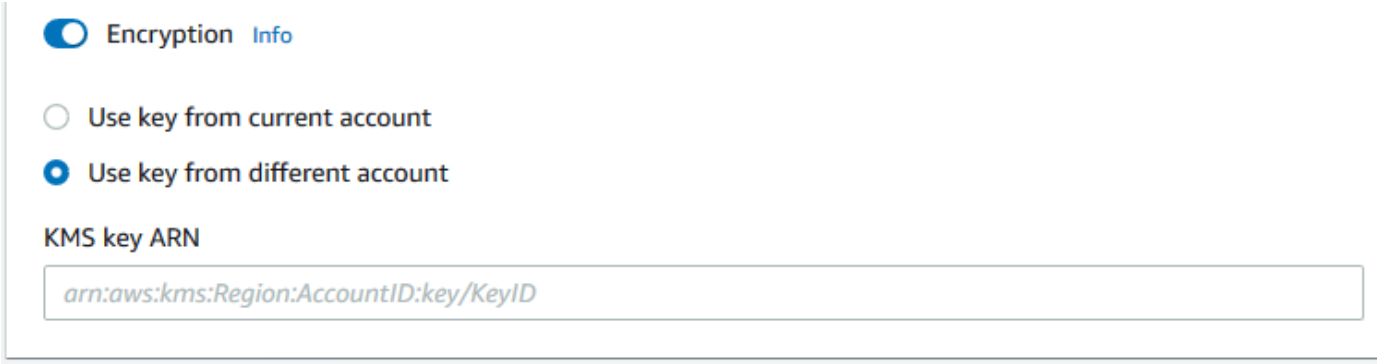
1. Sous Paramètres du Job, choisissez l'option de chiffrement du Job.



2. Choisissez si la clé gérée par le client KMS (CMK) provient du compte que vous utilisez actuellement ou d'un autre compte. Si vous souhaitez utiliser une clé du compte courant, choisissez l'alias de clé dans l'ID de clé KMS. Si vous utilisez une clé d'un autre compte, vous devez saisir l'ARN de la clé.

Pour activer le chiffrement des résultats en sortie

1. Sous Paramètres de sortie, choisissez l'option Chiffrement.



2. Choisissez si la clé gérée par le client (CMK) provient du compte que vous utilisez actuellement ou d'un autre compte. Si vous souhaitez utiliser une clé du compte actuel, choisissez l'ID de clé dans l'ID de clé KMS. Si vous utilisez une clé d'un autre compte, vous devez saisir l'ARN de la clé.

Si vous avez déjà configuré le chiffrement à l'aide de SSE-KMS sur vos documents d'entrée S3, cela peut vous apporter une sécurité supplémentaire. Toutefois, dans ce cas, le rôle IAM utilisé doit être `kms:Decrypt` autorisé à utiliser la clé KMS avec laquelle les documents d'entrée sont chiffrés. Pour de plus amples informations, veuillez consulter Autorisations requises pour utiliser le chiffrement KMS.

Chiffrement KMS avec opérations d'API

Toutes les opérations Amazon Comprehend `Start*` et `Create*` API prennent en charge les documents d'entrée cryptés KMS. `Describe*`et les opérations d'`List*`API renvoient `KmsKeyId` l'entrée `OutputDataConfig` si la tâche d'origine avait été `KmsKeyId` fournie en entrée. S'il n'a pas été fourni en entrée, il n'est pas renvoyé.

Cela peut être vu dans l'exemple de AWS CLI suivant utilisant l'StartEntitiesDetectionJobopération :

```
aws comprehend start-entities-detection-job \
     --region region \
     --data-access-role-arn "data access role arn" \
     --entity-recognizer-arn "entity recognizer arn" \
     --input-data-config "S3Uri=s3://Bucket Name/Bucket Path" \
     --job-name job name \
     --language-code en \
     --output-data-config "KmsKeyId=Output S3 KMS key ID" "S3Uri=s3://Bucket
 Name/Bucket Path/" \
     --volumekmskeyid "Volume KMS key ID"
```

> **Note**
>
> Cet exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Chiffrement par clé gérée par le client (CMK) avec opérations d'API

Les opérations d'API du modèle personnalisé Amazon Comprehend, `CreateEntityRecognizer` `CreateDocumentClassifierCreateEndpoint`, et prennent en charge le chiffrement à l'aide de clés gérées par le client via le. AWS CLI

Vous avez besoin d'une politique IAM pour autoriser un mandant à utiliser ou à gérer les clés gérées par le client. Ces clés sont spécifiées dans l'`Resource`élément de la déclaration de politique. Il est recommandé de limiter les clés gérées par le client à celles que les principaux doivent utiliser dans votre déclaration de politique.

L'exemple de AWS CLI suivant crée un outil de reconnaissance d'entités personnalisé avec chiffrement de modèle à l'aide de l'CreateEntityRecognizeropération :

```
aws comprehend create-entity-recognizer \
     --recognizer-name name \
     --data-access-role-arn data access role arn \
     --language-code en \
     --model-kms-key-id Model KMS Key ID \
     --input-data-config file:///path/input-data-config.json
```

> **Note**
>
> Cet exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition `aws:SourceAccount`globale `aws:SourceArn`et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qu'Amazon Comprehend accorde à un autre service à la ressource. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de stratégie.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Utilisation du compte source

L'exemple suivant montre comment utiliser la clé de contexte de condition `aws:SourceAccount` globale dans Amazon Comprehend.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
          "Service": "comprehend.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringEquals": {
             "aws:SourceAccount":"111122223333"
        }
    }
  }
}
```

Politique de confiance pour les terminaux des modèles chiffrés

Vous devez créer une politique de confiance pour créer ou mettre à jour un point de terminaison pour un modèle chiffré. Définissez la valeur `aws:SourceAccount` sur l'ID de votre compte. Si vous utilisez la `ArnEquals` condition, définissez la `aws:SourceArn` valeur sur l'ARN du point de terminaison.

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": "comprehend.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
               "aws:SourceAccount": "111122223333"
```

```
            },
            "ArnEquals": {
               "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier-endpoint/endpoint-name"
            }
        }
    }
  ]
}
```

Création d'un modèle personnalisé

Vous devez créer une politique de confiance pour créer un modèle personnalisé. Définissez la valeur `aws:SourceAccount` sur l'ID de votre compte. Si vous utilisez la `ArnEquals` condition, définissez la `aws:SourceArn` valeur sur l'ARN de la version du modèle personnalisé.

```
{
 "Version": "2012-10-17",
  "Statement": [
      {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
              "Service": "comprehend.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": "111122223333"
              },
              "ArnEquals": {
                  "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:
                            document-classifier/smallest-classifier-test/
version/version-name"
              }
          }
      }
  ]
}
```

Protégez les emplois en utilisant un Amazon Virtual Private Cloud

Amazon Comprehend utilise diverses mesures de sécurité pour garantir la sécurité de vos données grâce à nos conteneurs de tâches dans lesquels elles sont stockées pendant leur utilisation par Amazon Comprehend. Cependant, les conteneurs de tâches accèdent à des AWS ressources, telles que les compartiments Amazon S3 dans lesquels vous stockez des données et des artefacts de modèles, via Internet.

Pour contrôler l'accès à vos données, nous vous recommandons de créer un cloud privé virtuel (VPC) et de le configurer de manière à ce que les données et les conteneurs ne soient pas accessibles via Internet. Pour plus d'informations sur la création et la configuration d'un VPC, consultez Démarrer avec Amazon VPC dans le Guide de l'utilisateur Amazon VPC. L'utilisation d'un VPC permet de protéger vos données, car vous pouvez configurer votre VPC de manière à ce qu'il ne soit pas connecté à Internet. L'utilisation d'un VPC vous permet également de surveiller l'ensemble du trafic réseau entrant et sortant de nos conteneurs de tâches à l'aide des journaux de flux VPC. Pour plus d'informations, consultez la rubrique Journaux de flux VPC dans le Guide de l'utilisateur Amazon VPC.

Vous spécifiez la configuration de votre VPC lorsque vous créez une tâche, en spécifiant les sous-réseaux et les groupes de sécurité. Lorsque vous spécifiez les sous-réseaux et les groupes de sécurité, Amazon Comprehend crée des interfaces réseau élastiques ENIs () associées à vos groupes de sécurité dans l'un des sous-réseaux. ENIs permettez à nos conteneurs de tâches de se connecter aux ressources de votre VPC. Pour plus d'informations ENIs, consultez la section Elastic Network Interfaces dans le guide de l'utilisateur Amazon VPC.

> **Note**
>
> Pour les tâches, vous pouvez uniquement configurer des sous-réseaux avec un VPC de location par défaut dans lequel votre instance s'exécute sur du matériel partagé. Pour plus d'informations sur l'attribut de location pour VPCs, consultez la section Instances dédiées dans le guide de l' EC2 utilisateur Amazon.

Configuration d'une tâche pour l'accès à Amazon VPC

Pour spécifier des sous-réseaux et des groupes de sécurité dans votre VPC, utilisez `VpcConfig` le paramètre de requête de l'API applicable ou fournissez ces informations lorsque vous créez une tâche dans la console Amazon Comprehend. Amazon Comprehend utilise ces informations pour les

créer ENIs et les joindre à nos conteneurs de tâches. Ils ENIs fournissent à nos conteneurs de tâches une connexion réseau au sein de votre VPC qui n'est pas connectée à Internet.

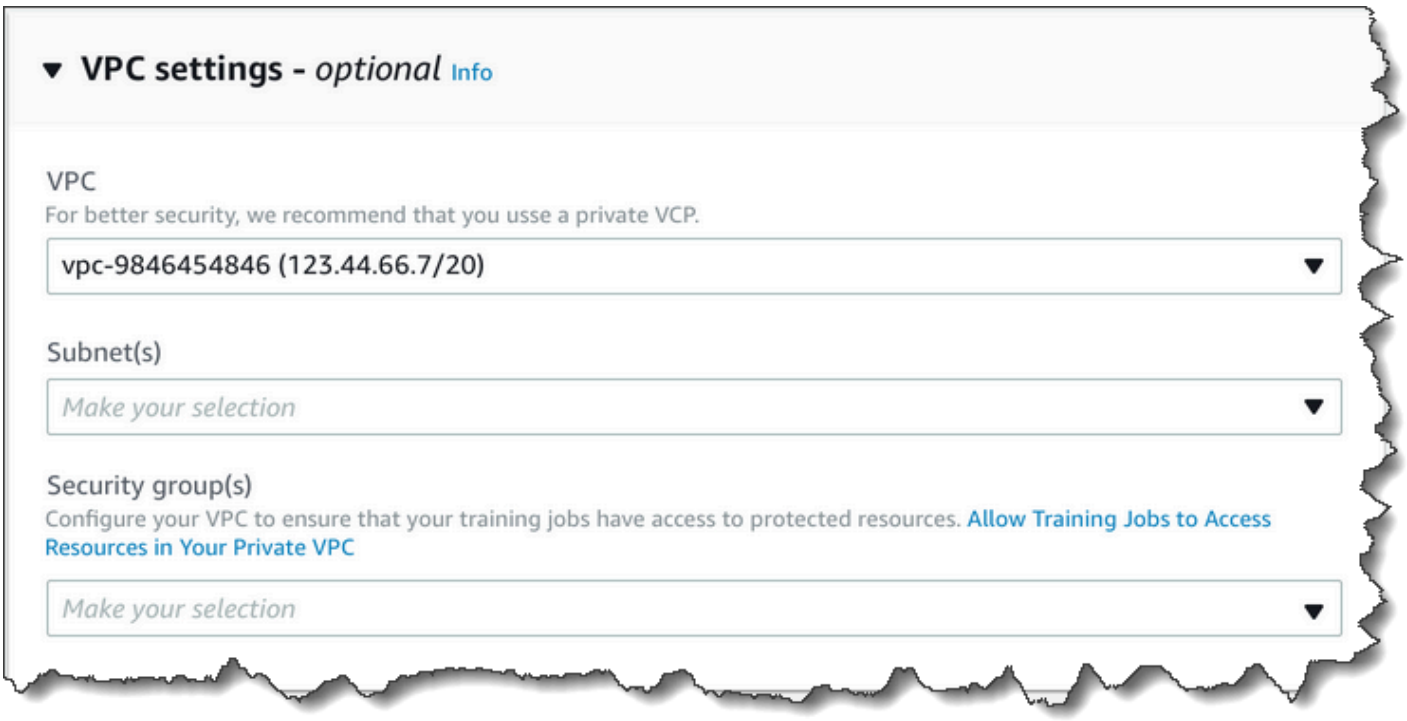
Les éléments suivants APIs contiennent le paramètre de `VpcConfig` demande :

- `Create*` APIs: `CreateDocumentClassifier`, `CreateEntityRecognizer`
- `Start*` APIs: `StartDocumentClassificationJob`, `StartDominantLanguageDetectionJob`, `StartEntitiesDetectionJob`, `StartKeyPhrasesDetectionJob`, `StartSentimentDetectionJob`, `StartTargetedSentimentDetectionJob`, `StartTopicsDetectionJob`

Voici un exemple du VpcConfig paramètre que vous incluez dans votre appel d'API :

```
"VpcConfig": {
      "SecurityGroupIds": [
          " sg-0123456789abcdef0"
          ],
      "Subnets": [
          "subnet-0123456789abcdef0",
          "subnet-0123456789abcdef1",
          "subnet-0123456789abcdef2"
          ]
      }
```

Pour configurer un VPC depuis la console Amazon Comprehend, choisissez les détails de configuration dans la section facultative VPC Settings lors de la création de la tâche.



Configurer votre VPC pour les tâches Amazon Comprehend

Lorsque vous configurez le VPC pour vos tâches Amazon Comprehend, suivez les instructions suivantes. Pour plus d'informations sur la configuration d'un VPC, consultez la section Utilisation des sous-réseaux VPCs et des sous-réseaux dans le guide de l'utilisateur Amazon VPC.

Assurez-vous que les sous-réseaux disposent d'un nombre suffisant d'adresses IP

Vos sous-réseaux VPC doivent avoir au moins deux adresses IP privées pour chaque instance d'une tâche. Pour plus d'informations, consultez la section relative au dimensionnement des VPC et des sous-réseaux dans le guide de l' IPv4utilisateur Amazon VPC.

Création d'un point de terminaison VPC Amazon S3

Si vous configurez votre VPC de telle sorte que les conteneurs de tâches n'aient pas accès à Internet, ils ne peuvent pas se connecter aux compartiments Amazon S3 contenant vos données, sauf si vous créez un point de terminaison VPC autorisant l'accès. En créant un point de terminaison VPC, vous autorisez les conteneurs de tâches à accéder à vos données pendant les tâches de formation et d'analyse.

Lorsque vous créez le point de terminaison VPC, configurez les valeurs suivantes :

- Sélectionnez la catégorie de service en tant que AWS Services

- Spécifiez le service en tant que `com.amazonaws.region.s3`
- Sélectionnez Gateway comme type de point de terminaison VPC

Si vous utilisez AWS CloudFormation pour créer le point de terminaison VPC, suivez la AWS CloudFormation VPCEndpointdocumentation. L'exemple suivant montre la VPCEndpointconfiguration dans un AWS CloudFormation modèle.

```
VpcEndpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action:
            - s3:GetObject
            - s3:PutObject
            - s3:ListBucket
            - s3:GetBucketLocation
            - s3:DeleteObject
            - s3:ListMultipartUploadParts
            - s3:AbortMultipartUpload
          Effect: Allow
          Resource:
            - "*"
          Principal: "*"
    RouteTableIds:
      - Ref: RouteTable
    ServiceName:
      Fn::Join:
        - ''
        - - com.amazonaws.
          - Ref: AWS::Region
          - ".s3"
    VpcId:
      Ref: VPC
```

Nous vous recommandons également de créer une politique personnalisée qui autorise uniquement les demandes provenant de votre VPC à accéder à vos compartiments S3. Pour de plus amples

informations, veuillez consulter Points de terminaison pour Amazon S3 dans le Guide de l'utilisateur Amazon VPC.

La stratégie suivante permet d'accéder aux compartiments S3. Modifiez cette politique pour autoriser l'accès uniquement aux ressources dont votre travail a besoin.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:DeleteObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "*"
        }
    ]
}
```

Utilisez les paramètres DNS par défaut pour la table de routage de votre point de terminaison, afin qu'Amazon S3 standard URLs (par exemple`http://s3-aws-region.amazonaws.com/amzn-s3-demo-bucket`) soit résolu. Si vous n'utilisez pas les paramètres DNS par défaut, assurez-vous que ceux URLs que vous utilisez pour spécifier l'emplacement des données dans vos tâches sont résolus en configurant les tables de routage des points de terminaison. Pour plus d'informations sur les tables de routage de point de terminaison d'un VPC, consultez Routage des points de terminaison de passerelle dans le Guide de l'utilisateur Amazon VPC.

La politique de point de terminaison par défaut permet aux utilisateurs d'installer des packages provenant des référentiels Amazon Linux et Amazon Linux 2 sur notre conteneur de tâches. Si vous ne voulez pas que les utilisateurs installent des packages à partir de ce référentiel, créez une politique de point de terminaison personnalisée qui refuse explicitement l'accès aux référentiels Amazon Linux et Amazon Linux 2. Comprehend lui-même n'a pas besoin de tels packages, il n'y aura donc aucun impact sur les fonctionnalités. Voici un exemple de politique qui refuse l'accès à ces référentiels :

```
{
    "Statement": [
      {
        "Sid": "AmazonLinuxAMIRepositoryAccess",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::packages.*.amazonaws.com/*",
            "arn:aws:s3:::repo.*.amazonaws.com/*"
        ]
      }
    ]
}

{
    "Statement": [
        { "Sid": "AmazonLinux2AMIRepositoryAccess",
          "Principal": "*",
          "Action": [
              "s3:GetObject"
              ],
          "Effect": "Deny",
          "Resource": [
              "arn:aws:s3:::amazonlinux.*.amazonaws.com/*"
              ]
         }
    ]
}
```

Autorisations pour **`DataAccessRole`**

Lorsque vous utilisez un VPC avec votre tâche d'analyse, les `Start*` opérations `DataAccessRole` used for the `Create*` et doivent également être autorisées à accéder au VPC à partir duquel les documents d'entrée et le compartiment de sortie sont accessibles.

La politique suivante fournit l'accès nécessaire aux opérations `DataAccessRole` utilisées pour les `Start*` opérations `Create*` et.

```
{
```

```
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",
              "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

Configuration du groupe de sécurité VPC

Dans le cas des tâches distribuées, vous devez autoriser la communication entre les différents conteneurs de tâches d'une même tâche. Pour ce faire, configurez une règle pour votre groupe de sécurité qui autorise les connexions entrantes entre les membres du même groupe de sécurité. Pour plus d'informations, consultez la section Règles des groupes de sécurité dans le guide de l'utilisateur Amazon VPC.

Connectez-vous à des ressources extérieures à votre VPC

Si vous configurez votre VPC de manière à ce qu'il n'ait pas accès à Internet, les tâches qui utilisent ce VPC n'ont pas accès à des ressources extérieures à votre VPC. Si vos tâches ont besoin d'accéder à des ressources extérieures à votre VPC, offrez l'accès à l'aide de l'une des options suivantes :

- Si votre travail nécessite l'accès à un AWS service prenant en charge les points de terminaison VPC d'interface, créez un point de terminaison pour vous connecter à ce service. Pour obtenir la liste des services qui prennent en charge les points de terminaison d'interface, veuillez consulter Points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC. Pour plus d'informations sur la création d'un point de terminaison VPC d'interface, consultez la section Points de terminaison VPC d'interface () dans le guide de AWS PrivateLink l'utilisateur Amazon VPC.

- Si votre tâche nécessite l'accès à un AWS service qui ne prend pas en charge les points de terminaison VPC d'interface ou à une ressource extérieure AWS, créez une passerelle NAT et configurez vos groupes de sécurité pour autoriser les connexions sortantes. Pour plus d'informations sur la configuration d'une passerelle NAT pour votre VPC, consultez Scénario 2 : VPC avec sous-réseaux publics et privés (NAT) dans le guide de l'utilisateur Amazon VPC.

Amazon Comprehend et interface avec les points de terminaison VPC ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et Amazon Comprehend en créant un point de terminaison VPC d'interface. Les points de terminaison de l'interface sont alimentés par AWS PrivateLinkune technologie qui vous permet d'accéder en privé à Amazon APIs Comprehend sans passerelle Internet, appareil NAT, connexion VPN ou connexion AWS Direct Connect. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec Amazon Comprehend. APIs Le trafic entre votre VPC et Amazon Comprehend ne quitte pas le réseau Amazon.

Chaque point de terminaison d'interface est représenté par une ou plusieurs interfaces réseau Elastic dans vos sous-réseaux.

Pour de plus amples informations, consultez Points de terminaison VPC (AWS PrivateLink) dans le Guide de l'utilisateur Amazon VPC.

Considérations relatives aux points de terminaison Amazon Comprehend VPC

Avant de configurer un point de terminaison VPC d'interface pour Amazon Comprehend, assurez-vous de consulter les propriétés et les limites du point de terminaison d'interface dans le guide de l'utilisateur Amazon VPC.

Les points de terminaison Amazon Comprehend ne sont pas disponibles dans toutes les zones de disponibilité d'une région. Lorsque vous créez le point de terminaison, utilisez la commande suivante pour répertorier les zones de disponibilité.

```
aws ec2 describe-vpc-endpoint-services \
  --service-names com.amazonaws.us-west-2.comprehend
```

Amazon Comprehend permet d'appeler toutes ses actions d'API depuis votre VPC.

Création d'un point de terminaison VPC d'interface pour Amazon Comprehend

Vous pouvez créer un point de terminaison VPC pour le service Amazon Comprehend à l'aide de la console Amazon VPC ou du (). AWS Command Line Interface AWS CLI Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Créez un point de terminaison VPC pour Amazon Comprehend en utilisant le nom de service suivant :

- com.amazonaws. *region*.comprendre

Si vous activez le DNS privé pour le point de terminaison, vous pouvez envoyer des demandes d'API à Amazon Comprehend en utilisant son nom DNS par défaut pour la région, par exemple,. *comprehend.us-east-1.amazonaws.com*

Pour plus d'informations, consultez Accès à un service via un point de terminaison d'interface dans le Guide de l'utilisateur Amazon VPC.

Création d'une politique de point de terminaison VPC pour Amazon Comprehend

Vous pouvez associer une politique de point de terminaison à votre point de terminaison VPC qui contrôle l'accès à Amazon Comprehend. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Exemple : politique de point de terminaison VPC pour les actions Amazon Comprehend

Voici un exemple de politique de point de terminaison pour Amazon Comprehend. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès à l'`DetectEntities`action Amazon Comprehend à tous les principaux sur toutes les ressources.

```
{
   "Statement":[
```

```
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "comprehend:DetectEntities"
         ],
         "Resource":"*"
      }
   ]
}
```

Identity and Access Management pour Amazon Comprehend

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources Amazon Comprehend. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Comment Amazon Comprehend fonctionne avec IAM
- Exemples de politiques basées sur l'identité pour Amazon Comprehend
- AWS politiques gérées pour Amazon Comprehend
- Résolution des problèmes d'identité et d'accès à Amazon Comprehend

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Amazon Comprehend.

Utilisateur du service : si vous utilisez le service Amazon Comprehend pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités

d'Amazon Comprehend pour effectuer votre travail, il se peut que vous ayez besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne parvenez pas à accéder à une fonctionnalité d'Amazon Comprehend, consultez. Résolution des problèmes d'identité et d'accès à Amazon Comprehend

Administrateur du service — Si vous êtes responsable des ressources Amazon Comprehend au sein de votre entreprise, vous avez probablement un accès complet à Amazon Comprehend. C'est à vous de déterminer les fonctionnalités et les ressources d'Amazon Comprehend auxquelles les utilisateurs de vos services doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser l'IAM avec Amazon Comprehend, consultez. Comment Amazon Comprehend fonctionne avec IAM

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à Amazon Comprehend. Pour consulter des exemples de politiques basées sur l'identité Amazon Comprehend que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour Amazon Comprehend

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez AWS Signature Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant des informations d'identification d'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez

vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM.

Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations SCPs, voir Politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.
- Politiques de contrôle des ressources (RCPs) : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les

autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.

- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment Amazon Comprehend fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon Comprehend, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon Comprehend.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon Comprehend

Fonctionnalité IAM	Assistance Amazon Comprehend
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui

Fonctionnalité IAM	Assistance Amazon Comprehend
ACLs	Non
ABAC (identifications dans les politiques)	Partielle
Informations d’identification temporaires	Oui
Transmission des sessions d’accès (FAS)	Oui
Rôles de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont Amazon Comprehend et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon Comprehend

Prend en charge les politiques basées sur l’identité : oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Définition d’autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Exemples de politiques basées sur l'identité pour Amazon Comprehend

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Comprehend, consultez. Exemples de politiques basées sur l'identité pour Amazon Comprehend

Politiques basées sur les ressources au sein d'Amazon Comprehend

Prend en charge les politiques basées sur les ressources : oui

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal intercompte à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Le service Amazon Comprehend ne prend en charge qu'un seul type de politique basée sur les ressources (une politique de modèle personnalisée), qui est attachée à un modèle personnalisé. Cette politique définit les autres comptes qui peuvent utiliser le modèle personnalisé.

Pour savoir comment associer une politique basée sur les ressources à un modèle personnalisé, consultez. Politiques basées sur les ressources pour les modèles personnalisés

Actions politiques pour Amazon Comprehend

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions Amazon Comprehend, consultez la section Actions définies par Amazon Comprehend dans le Service Authorization Reference.

Les actions politiques dans Amazon Comprehend utilisent le préfixe suivant avant l'action :

```
comprehend
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "comprehend:DetectSentiment",
      "comprehend:ClassifyDocument"
         ]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "comprehend:Describe*"
```

N'utilisez pas de caractères génériques pour spécifier toutes les actions d'un service. Utilisez la meilleure pratique qui consiste à accorder le moindre privilège lorsque vous spécifiez les autorisations dans une politique.

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Comprehend, consultez. Exemples de politiques basées sur l'identité pour Amazon Comprehend

Ressources relatives aux politiques pour Amazon Comprehend

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de ressources Amazon Comprehend et leurs caractéristiques ARNs, consultez la section Ressources définies par Amazon Comprehend dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon Comprehend.

Clés de conditions de politique pour Amazon Comprehend

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition d'Amazon Comprehend, consultez la section Clés de condition pour Amazon Comprehend dans le Service Authorization Reference. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon Comprehend.

Pour consulter des exemples de politiques basées sur l'identité d'Amazon Comprehend, consultez. Exemples de politiques basées sur l'identité pour Amazon Comprehend

ACLs dans Amazon Comprehend

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec Amazon Comprehend

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur le balisage des ressources Amazon Comprehend, consultez. Balisage de vos ressources

Utilisation d'informations d'identification temporaires avec Amazon Comprehend

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Passage d'un rôle utilisateur à un rôle IAM (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires

au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Sessions d'accès transféré pour Amazon Comprehend

Prend en charge les sessions d'accès direct (FAS) : oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour Amazon Comprehend

Prend en charge les rôles de service : oui

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon Comprehend. Modifiez les rôles de service uniquement lorsque Amazon Comprehend fournit des instructions à cet effet.

Pour utiliser les opérations asynchrones d'Amazon Comprehend, vous devez autoriser Amazon Comprehend à accéder au compartiment Amazon S3 qui contient votre collection de documents. Pour ce faire, vous devez créer un rôle d'accès aux données dans votre compte avec une politique de confiance visant à faire confiance au principal du service Amazon Comprehend.

Pour un exemple de stratégie, voir Autorisations basées sur les rôles requises pour les opérations asynchrones

Rôles liés à un service pour Amazon Comprehend

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour Amazon Comprehend

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources Amazon Comprehend. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon Comprehend, y compris le format de chaque type de ressource, consultez la section Actions, ressources et clés de condition pour Amazon Comprehend dans le Service Authorization Reference. ARNs

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console Amazon Comprehend
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Autorisations requises pour effectuer des actions d'analyse de documents
- Autorisations requises pour utiliser le chiffrement KMS
- AWS politiques gérées (prédéfinies) pour Amazon Comprehend

- Autorisations basées sur les rôles requises pour les opérations asynchrones
- Autorisations permettant d'autoriser toutes les actions Amazon Comprehend
- Autorisations permettant d'autoriser les actions de modélisation de sujets
- Autorisations requises pour une tâche d'analyse asynchrone personnalisée

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Amazon Comprehend dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les

bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console Amazon Comprehend

Pour accéder à la console Amazon Comprehend, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon Comprehend présentes dans votre. Compte AWS Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour obtenir des autorisations minimales sur la console Amazon Comprehend, vous pouvez associer la politique *ComprehendReadOnly* AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

Pour utiliser la console Amazon Comprehend, vous devez également disposer d'autorisations pour effectuer les actions décrites dans la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
      "Action": [
          "iam:ListRoles",
```

```
            "iam:GetRole",
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
    ]
    }
```

La console Amazon Comprehend a besoin de ces autorisations supplémentaires pour les raisons suivantes :

- `iam`autorisations permettant de répertorier les rôles IAM disponibles pour votre compte.
- `s3`autorisations d'accès aux compartiments et aux objets Amazon S3 contenant les données pour la modélisation de sujets.

Lorsque vous créez une tâche par lots asynchrone ou une tâche de modélisation de rubriques à l'aide de la console, vous pouvez demander à la console de créer un rôle IAM pour votre tâche. Pour créer un rôle IAM, les utilisateurs doivent disposer des autorisations supplémentaires suivantes pour créer des rôles et des politiques IAM, et pour associer des politiques aux rôles :

```
{
  "Version": "2012-10-17",
  "Statement":
    [
      {
        "Action":
          [
          "iam:CreateRole",
          "iam:CreatePolicy",
          "iam:AttachRolePolicy"
          ],
        "Effect": "Allow",
        "Resource": "*"
      },
      {
        "Action":
          [
          "iam:PassRole"
```

```
          ],
        "Effect": "Allow",
        "Resource": "arn:aws:iam::*:role/*Comprehend*"
      }
    ]
  }
```

La console Amazon Comprehend a besoin de ces autorisations supplémentaires pour les raisons suivantes :

- `iam`autorisations pour créer des rôles et des politiques et pour associer des rôles et des politiques. L'`iam:PassRole`action permet à la console de transmettre le rôle à Amazon Comprehend.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
```

```
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Autorisations requises pour effectuer des actions d'analyse de documents

L'exemple de politique suivant accorde des autorisations pour utiliser les actions d'analyse de documents Amazon Comprehend :

```
{
 "Version": "2012-10-17",
 "Statement": [{
    "Sid": "AllowDetectActions",
    "Effect": "Allow",
    "Action": [
              "comprehend:DetectEntities",
              "comprehend:DetectKeyPhrases",
              "comprehend:DetectDominantLanguage",
              "comprehend:DetectSentiment",
              "comprehend:DetectTargetedSentiment",
              "comprehend:DetectSyntax",
              "textract:DetectDocumentText",
              "textract:AnalyzeDocument"
           ],
    "Resource": "*"
    }
 ]
}
```

La politique comporte une déclaration qui autorise l'utilisation des `DetectSyntax` actions `DetectEntitiesDetectKeyPhrases`, `DetectDominantLanguageDetectTargetedSentiment`,`DetectSentiment`,, et. La déclaration de politique accorde également l'autorisation d'utiliser deux méthodes d'API Amazon Textract. Amazon Comprehend fait appel à ces méthodes pour extraire du texte à partir de fichiers image et

de documents PDF numérisés. Vous pouvez supprimer ces autorisations pour les utilisateurs qui n'exécutent jamais d'inférence personnalisée pour ces types de fichiers d'entrée.

Un utilisateur appliquant cette politique ne serait pas en mesure d'effectuer des actions par lots ou des actions asynchrones dans votre compte.

La stratégie ne spécifie pas l'élément `Principal`, car vous ne spécifiez pas le mandataire qui obtient l'autorisation dans une stratégie basée sur une identité. Quand vous attachez une stratégie à un utilisateur, l'utilisateur est le mandataire implicite. Lorsque vous attachez une politique d'autorisation à un rôle IAM, le principal identifié dans la politique d'approbation de ce rôle obtient les autorisations.

Pour consulter un tableau présentant toutes les actions de l'API Amazon Comprehend et les ressources auxquelles elles s'appliquent, consultez la section Actions, ressources et clés de condition pour Amazon Comprehend dans le Service Authorization Reference.

Autorisations requises pour utiliser le chiffrement KMS

Pour utiliser pleinement Amazon Key Management Service (KMS) pour le chiffrement des données et des tâches dans le cadre d'une tâche asynchrone, vous devez accorder des autorisations pour les actions décrites dans la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Action": [
            "kms:CreateGrant"
        ],
        "Effect": "Allow",
        "Resource": "*"
    },
    {
        "Action": [
            "kms:Decrypt",
            "kms:GenerateDatakey"
        ],
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": [
```

```
                    "s3.region.amazonaws.com"
                ]
            }
        }
    }
  ]
}
```

Lorsque vous créez une tâche asynchrone avec Amazon Comprehend, vous utilisez des données d'entrée stockées sur Amazon S3. Avec S3, vous avez la possibilité de chiffrer vos données stockées, qui sont chiffrées par S3, et non par Amazon Comprehend. Nous pouvons déchiffrer et lire ces données d'entrée cryptées si vous `kms:Decrypt` autorisez la clé avec laquelle les données d'entrée d'origine ont été cryptées au rôle d'accès aux données utilisé par la tâche Amazon Comprehend.

Vous avez également la possibilité d'utiliser des clés gérées par le client (CMK) KMS pour chiffrer les résultats de sortie sur S3, ainsi que le volume de stockage utilisé pendant le traitement des tâches. Dans ce cas, vous pouvez utiliser la même clé KMS pour les deux types de chiffrement, mais cela n'est pas nécessaire. Des champs distincts sont disponibles lors de la création de la tâche pour spécifier les clés de chiffrement de sortie et de chiffrement de volume. Vous pouvez même utiliser une clé KMS provenant d'un autre compte.

Lors de l'utilisation du chiffrement KMS, une `kms:CreateGrant` autorisation est requise pour le chiffrement des volumes et une `kms:GenerateDataKey` autorisation est requise pour le chiffrement des données de sortie. Pour lire une entrée cryptée (comme lorsque les données d'entrée sont déjà chiffrées par Amazon S3), une `kms:Decrypt` autorisation est requise. Le rôle IAM doit accorder ces autorisations selon les besoins. Toutefois, si la clé provient d'un compte différent de celui actuellement utilisé, la politique de clé KMS associée à cette clé KMS doit également accorder ces autorisations au rôle d'accès aux données pour la tâche.

AWS politiques gérées (prédéfinies) pour Amazon Comprehend

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Ces politiques AWS gérées accordent les autorisations nécessaires pour les cas d'utilisation courants afin que vous puissiez éviter d'avoir à rechercher les autorisations nécessaires. Pour plus d’informations, consultez Politiques gérées par AWS dans le Guide de l’utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à Amazon Comprehend :

- ComprehendFullAccess— Accorde un accès complet aux ressources d'Amazon Comprehend, y compris l'exécution de tâches de modélisation de sujets. Inclut l'autorisation de répertorier et d'obtenir des rôles IAM.
- ComprehendReadOnly— Octroie l'autorisation d'exécuter toutes les actions Amazon Comprehend à l'exception de `StartDominantLanguageDetectionJobStartEntitiesDetectionJob,,StartKeyPhrasesDetec StartSentimentDetectionJobStartTargetedSentimentDetectionJob`, et. `StartTopicsDetectionJob`

Vous devez appliquer la politique supplémentaire suivante à tout utilisateur qui utilisera Amazon Comprehend :

```
{
    "Version": "2012-10-17",
    "Statement":
      [
        {
          "Action":
            [
              "iam:PassRole"
            ],
          "Effect": "Allow",
          "Resource": "arn:aws:iam::*:role/*Comprehend*"
        }
      ]
  }
```

Vous pouvez consulter les politiques d'autorisations gérées en vous connectant à la console IAM et en y recherchant des politiques spécifiques.

Ces politiques fonctionnent lorsque vous utilisez AWS SDKs la AWS CLI.

Vous pouvez également créer vos propres politiques IAM personnalisées pour autoriser les actions et les ressources Amazon Comprehend. Vous pouvez associer ces politiques personnalisées aux utilisateurs, groupes ou rôles qui nécessitent ces autorisations.

Autorisations basées sur les rôles requises pour les opérations asynchrones

Pour utiliser les opérations asynchrones d'Amazon Comprehend, vous devez accorder à Amazon Comprehend l'accès au compartiment Amazon S3 qui contient votre collection de documents.

Pour ce faire, vous devez créer un rôle d'accès aux données dans votre compte avec une politique de confiance visant à faire confiance au principal du service Amazon Comprehend. Pour plus d'informations sur la création d'un rôle, consultez la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le guide de l'utilisateur d'AWS Identity and Access Management.

Voici un exemple de politique de confiance pour le rôle que vous créez. Pour éviter toute confusion chez les adjoints, vous limitez la portée de l'autorisation en utilisant une ou plusieurs clés contextuelles de conditions globales. Définissez la valeur `aws:SourceAccount` sur l'ID de votre compte. Si vous utilisez la `ArnEquals` condition, définissez la `aws:SourceArn` valeur sur l'ARN de la tâche. Utilisez un caractère générique pour le numéro de tâche dans l'ARN, car Amazon Comprehend génère ce numéro dans le cadre de la création d'emploi.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "comprehend.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "111122223333"
          },
          "ArnEquals": {
            "aws:SourceArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-
detection-job/*"
          }
        }
      }
    ]
 }
```

Après avoir créé le rôle, créez une politique d'accès pour ce rôle. Cela devrait accorder à Amazon S3 `GetObject` et `ListBucket` des autorisations au compartiment Amazon S3 qui contient vos données d'entrée, ainsi que l'`PutObject`autorisation Amazon S3 à votre compartiment de données de sortie Amazon S3.

Autorisations permettant d'autoriser toutes les actions Amazon Comprehend

Une fois inscrit AWS, vous créez un utilisateur administrateur chargé de gérer votre compte, notamment de créer des utilisateurs et de gérer leurs autorisations.

Vous pouvez choisir de créer un utilisateur autorisé à effectuer toutes les actions Amazon Comprehend (considérez cet utilisateur comme un administrateur spécifique à un service) afin de travailler avec Amazon Comprehend. Vous pouvez alors lier la stratégie d'autorisations suivante à cet utilisateur.

```
{
  "Version": "2012-10-17",
  "Statement":
    [
      {
        "Sid": "AllowAllComprehendActions",
        "Effect": "Allow",
        "Action":
        [
            "comprehend:*",
            "iam:ListRoles",
            "iam:GetRole",
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation",
            "iam:CreateRole",
            "iam:CreatePolicy",
            "iam:AttachRolePolicy",
            "kms:CreateGrant",
            "kms:Decrypt",
            "kms:GenerateDatakey"
      ],
      "Resource": "*"
    },
    {
      "Action":
        [
          "iam:PassRole"
        ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/*Comprehend*"
    }
  ]
```

```
}
```

Ces autorisations peuvent être modifiées en ce qui concerne le chiffrement de la manière suivante :

- Pour permettre à Amazon Comprehend d'analyser des documents stockés dans un compartiment S3 chiffré, le rôle IAM doit disposer de cette autorisation. `kms:Decrypt`
- Pour permettre à Amazon Comprehend de chiffrer des documents stockés sur un volume de stockage attaché à l'instance de calcul qui traite le travail d'analyse, le rôle IAM doit disposer de l'autorisation. `kms:CreateGrant`
- Pour permettre à Amazon Comprehend de chiffrer les résultats de sortie dans son compartiment S3, le rôle IAM doit disposer de cette autorisation. `kms:GenerateDataKey`

Autorisations permettant d'autoriser les actions de modélisation de sujets

La politique d'autorisation suivante accorde aux utilisateurs les autorisations nécessaires pour effectuer les opérations de modélisation des rubriques Amazon Comprehend.

```
{
  "Version": "2012-10-17",
  "Statement": [{
  "Sid": "AllowTopicModelingActions",
  "Effect": "Allow",
  "Action": [
            "comprehend:DescribeTopicsDetectionJob",
            "comprehend:ListTopicsDetectionJobs",
            "comprehend:StartTopicsDetectionJob",
         ],
        "Resource": "*"
        ]
    }
  ]
  }
```

Autorisations requises pour une tâche d'analyse asynchrone personnalisée

> **Important**
>
> Si vous avez une politique IAM qui restreint l'accès aux modèles, vous ne serez pas en mesure de terminer une tâche d'inférence avec un modèle personnalisé. Votre politique

IAM doit être mise à jour pour inclure une ressource générique pour une tâche d'analyse asynchrone personnalisée.

Si vous utilisez le StartDocumentClassificationJob et StartEntitiesDetectionJob APIs, vous devez mettre à jour votre politique IAM, sauf si vous utilisez actuellement des caractères génériques comme ressources. Si vous StartEntitiesDetectionJobutilisez un modèle préentraîné, cela ne vous concerne pas et vous n'avez aucune modification à apporter.

L'exemple de politique suivant contient une référence obsolète.

```
{
    "Action": [
        "comprehend:StartDocumentClassificationJob",
        "comprehend:StartEntitiesDetectionJob",
    ],
    "Resource": [
        "arn:aws:comprehend:us-east-1:123456789012:document-classifier/myClassifier",
        "arn:aws:comprehend:us-east-1:123456789012:entity-recognizer/myRecognizer"
    ],
    "Effect": "Allow"
}
```

Il s'agit de la politique mise à jour que vous devez utiliser pour exécuter StartDocumentClassificationJob correctement et. StartEntitiesDetectionJob

```
{
    "Action": [
        "comprehend:StartDocumentClassificationJob",
        "comprehend:StartEntitiesDetectionJob",
    ],
    "Resource": [
        "arn:aws:comprehend:us-east-1:123456789012:document-classifier/myClassifier",
        "arn:aws:comprehend:us-east-1:123456789012:document-classification-job/*",
        "arn:aws:comprehend:us-east-1:123456789012:entity-recognizer/myRecognizer",
        "arn:aws:comprehend:us-east-1:123456789012:entities-detection-job/*"
    ],
    "Effect": "Allow"
}
```

AWS politiques gérées pour Amazon Comprehend

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent des cas d'utilisation courants et sont disponibles dans votre Compte AWS. Pour plus d'informations sur les politiques AWS gérées, voir les politiques AWS gérées dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique ReadOnlyAccess AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page politiques gérées par AWS pour les fonctions de tâche dans le Guide de l'utilisateur IAM.

AWS politique gérée : ComprehendFullAccess

Cette politique accorde un accès complet aux ressources d'Amazon Comprehend, y compris l'exécution de tâches de modélisation de sujets. Cette politique accorde également des autorisations de liste et d'obtention pour les buckets Amazon S3 et les rôles IAM.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "comprehend:*",
                "iam:GetRole",
                "iam:ListRoles",
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politique gérée : ComprehendReadOnly

Cette politique accorde des autorisations en lecture seule pour exécuter toutes les actions Amazon Comprehend, à l'exception des suivantes :

- `StartDominantLanguageDetectionJob`
- `StartEntitiesDetectionJob`
- `StartKeyPhrasesDetectionJob`
- `StartSentimentDetectionJob`
- `StartTargetedSentimentDetectionJob`
- `StartTopicsDetectionJob`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "comprehend:BatchDetectDominantLanguage",
                "comprehend:BatchDetectEntities",
                "comprehend:BatchDetectKeyPhrases",
                "comprehend:BatchDetectSentiment",
                "comprehend:BatchDetectSyntax",
                "comprehend:ClassifyDocument",
                "comprehend:ContainsPiiEntities",
```

```
                "comprehend:DescribeDocumentClassificationJob",
                "comprehend:DescribeDocumentClassifier",
                "comprehend:DescribeDominantLanguageDetectionJob",
                "comprehend:DescribeEndpoint",
                "comprehend:DescribeEntitiesDetectionJob",
                "comprehend:DescribeEntityRecognizer",
                "comprehend:DescribeKeyPhrasesDetectionJob",
                "comprehend:DescribePiiEntitiesDetectionJob",
                "comprehend:DescribeResourcePolicy",
                "comprehend:DescribeSentimentDetectionJob",
                "comprehend:DescribeTargetedSentimentDetectionJob",
                "comprehend:DescribeTopicsDetectionJob",
                "comprehend:DetectDominantLanguage",
                "comprehend:DetectEntities",
                "comprehend:DetectKeyPhrases",
                "comprehend:DetectPiiEntities",
                "comprehend:DetectSentiment",
                "comprehend:DetectSyntax",
                "comprehend:ListDocumentClassificationJobs",
                "comprehend:ListDocumentClassifiers",
                "comprehend:ListDocumentClassifierSummaries",
                "comprehend:ListDominantLanguageDetectionJobs",
                "comprehend:ListEndpoints",
                "comprehend:ListEntitiesDetectionJobs",
                "comprehend:ListEntityRecognizers",
                "comprehend:ListEntityRecognizerSummaries",
                "comprehend:ListKeyPhrasesDetectionJobs",
                "comprehend:ListPiiEntitiesDetectionJobs",
                "comprehend:ListSentimentDetectionJobs",
                "comprehend:ListTargetedSentimentDetectionJobs",
                "comprehend:ListTagsForResource",
                "comprehend:ListTopicsDetectionJobs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Amazon Comprehend met à jour les politiques gérées AWS

Consultez les informations relatives aux mises à jour des politiques AWS gérées pour Amazon Comprehend depuis que ce service a commencé à suivre ces modifications. Pour recevoir des

alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document Amazon Comprehend.

Modification	Description	Date
ComprehendReadOnly – Mise à jour d'une stratégie existante	Amazon Comprehend autorise désormais les `comprehend:ListTargetedSentimentDetectionJobs` actions `comprehend:DescribeTargetedSentimentDetectionJob` et les actions prévues dans la politique ComprehendReadOnly	30 mars 2022
ComprehendReadOnly – Mise à jour d'une politique existante	Amazon Comprehend autorise désormais l'`comprehend:DescribeResourcePolicy` action prévue dans la politique ComprehendReadOnly	2 févr. 2022
ComprehendReadOnly – Mise à jour d'une politique existante	Amazon Comprehend autorise désormais les `ListEntityRecognizerSummaries` actions `ListDocumentClassifierSummaries` et les actions prévues dans la politique ComprehendReadOnly	21 septembre 2021
ComprehendReadOnly – Mise à jour d'une politique existante	Amazon Comprehend autorise désormais l'`ContainsPIIEntities` action prévue dans la politique ComprehendReadOnly	26 mars 2021

Modification	Description	Date
Amazon Comprehend a commencé à suivre les modifications	Amazon Comprehend a commencé à suivre les modifications apportées à ses politiques AWS gérées.	1er mars 2021

Résolution des problèmes d'identité et d'accès à Amazon Comprehend

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon Comprehend et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Amazon Comprehend
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Amazon Comprehend

Je ne suis pas autorisé à effectuer une action dans Amazon Comprehend

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `comprehend:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 comprehend:GetWidget on resource: my-example-widget
```

Dans ce cas, la stratégie de Mateo doit être mise à jour pour l'autoriser à accéder à la ressource *`my-example-widget`* à l'aide de l'action `comprehend:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon Comprehend.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon Comprehend. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Amazon Comprehend

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si Amazon Comprehend prend en charge ces fonctionnalités, consultez. Comment Amazon Comprehend fonctionne avec IAM
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Journalisation des appels d'API Amazon Comprehend avec AWS CloudTrail

Amazon Comprehend est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans Amazon Comprehend. CloudTrail capture les appels d'API pour Amazon Comprehend sous forme d'événements. Les appels capturés incluent des appels provenant de la console Amazon Comprehend et des appels de code vers les opérations de l'API Amazon Comprehend. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour Amazon Comprehend. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon Comprehend, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le guide de AWS CloudTrail l'utilisateur.

Informations sur Amazon Comprehend dans CloudTrail

CloudTrail est activé sur votre compte Compte AWS lorsque vous créez le compte. Lorsqu'une activité événementielle prise en charge se produit dans Amazon Comprehend, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre entreprise Compte AWS, y compris des événements pour Amazon Comprehend, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal de suivi consigne les événements de toutes les régions dans la partition AWS , et il livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Amazon Comprehend prend en charge l'enregistrement des actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux :

- BatchDetectDominantLanguage
- BatchDetectEntities
- BatchDetectKeyPhrases
- BatchDetectSentiment
- BatchDetectSyntax
- ClassifyDocument
- CreateDocumentClassifier
- CreateEndpoint
- CreateEntityRecognizer
- DeleteDocumentClassifier
- DeleteEndpoint
- DeleteEntityRecognizer
- DescribeDocumentClassificationJob
- DescribeDocumentClassifier
- DescribeDominantLanguageDetectionJob

- DescribeEndpoint
- DescribeEntitiesDetectionJob
- DescribeEntityRecognizer
- DescribeKeyPhrasesDetectionJob
- DescribePiiEntitiesDetectionJob
- DescribeSentimentDetectionJob
- DescribeTargetedSentimentDetectionJob
- DescribeTopicsDetectionJob
- DetectDominantLanguage
- DetectEntities
- DetectKeyPhrases
- DetectPiiEntities
- DetectSentiment
- DetectSyntax
- ListDocumentClassificationJobs
- ListDocumentClassifiers
- ListDominantLanguageDetectionJobs
- ListEndpoints
- ListEntitiesDetectionJobs
- ListEntityRecognizers
- ListKeyPhrasesDetectionJobs
- ListPiiEntitiesDetectionJobs
- ListSentimentDetectionJobs
- ListTargetedSentimentDetectionJobs
- ListTagsForResource
- ListTopicsDetectionJobs
- StartDocumentClassificationJob
- StartDominantLanguageDetectionJob
- StartEntitiesDetectionJob
- StartKeyPhrasesDetectionJob

- StartPiiEntitiesDetectionJob
- StartSentimentDetectionJob
- StartTargetedSentimentDetectionJob
- StartTopicsDetectionJob
- StopDominantLanguageDetectionJob
- StopEntitiesDetectionJob
- StopKeyPhrasesDetectionJob
- StopPiiEntitiesDetectionJob
- StopSentimentDetectionJob
- StopTargetedSentimentDetectionJob
- StopTrainingDocumentClassifier
- StopTrainingEntityRecognizer
- TagResource
- UntagResource
- UpdateEndpoint

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Exemple : entrées du fichier journal Amazon Comprehend

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent

pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`ClassifyDocument`action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AROAICFHPEXAMPLE",
        "arn": "arn:aws:iam::12345678910:user/myadmin2",
        "accountId": "12345678910",
        "accessKeyId": "ASIA3VZEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-10-19T14:22:09Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-10-19T17:31:20Z",
    "eventSource": "comprehend.amazonaws.com",
    "eventName": "ClassifyDocument",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "3.21.185.237",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0)
 Gecko/20100101 Firefox/115.0",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "fd916e66-caac-46c9-a1fc-81a0ef33e61b",
    "eventID": "535ca22b-b3a3-4c13-b2c5-bf51ab082794",
    "readOnly": false,
    "resources": [
        {
            "accountId": "12345678910",
            "type": "AWS::Comprehend::DocumentClassifierEndpoint",
            "ARN": "arn:aws:comprehend:us-east-2:12345678910:document-classifier-
endpoint/endpointExample"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "12345678910"
```

```
}
```

Validation de conformité pour Amazon Comprehend

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon Comprehend dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications PCI, FedRAMP, HIPAA et autres. Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, consultez la section Téléchargement de rapports dans AWS Artifact.

Lorsque vous utilisez Amazon Comprehend, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources relatives à la conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Config— Ce AWS service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformité. Pour obtenir des informations générales, consultez Programmes de conformitéAWS.

La résilience dans Amazon Comprehend

L'infrastructure AWS mondiale est construite autour de Région AWS s et de zones de disponibilité. Région AWS s fournissent plusieurs zones de disponibilité physiquement séparées et isolées,

connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur les Région AWS zones de disponibilité et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Sécurité de l'infrastructure dans Amazon Comprehend

En tant que service géré, Amazon Comprehend est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon Comprehend via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Consignes et quotas

Sauf indication contraire, les quotas Amazon Comprehend sont définis par région. Vous pouvez demander une augmentation des quotas ajustables si cela est nécessaire pour vos applications. Pour plus d'informations sur les quotas et pour demander une augmentation de quota, consultez AWS Service Quotas.

Rubriques

- Régions prises en charge
- Quotas pour les modèles intégrés
- Quotas pour les modèles personnalisés
- Quotas pour les volants

Régions prises en charge

Amazon Comprehend est disponible dans les régions suivantes : AWS

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- AWS GovCloud (US-Ouest)

Par défaut, Amazon Comprehend fournit toutes les opérations d'API dans chacune des régions prises en charge. Pour les exceptions, voirTraitement des documents.

Pour plus d'informations sur les points de terminaison d'API, consultez Amazon Comprehend Regions and Endpoints dans le manuel Amazon Web Services General Reference.

Pour consulter les quotas actuels dans une région, ou pour demander des augmentations de quotas pour des quotas ajustables, ouvrez la console Service Quotas.

Quotas pour les modèles intégrés

Amazon Comprehend fournit des modèles intégrés qui vous permettent d'analyser des documents texte UTF-8. Amazon Comprehend fournit des opérations synchrones et asynchrones qui utilisent les modèles intégrés.

Rubriques

- Analyse en temps réel (synchrone)
- Analyse asynchrone

Analyse en temps réel (synchrone)

Cette section décrit les quotas liés à l'analyse en temps réel à l'aide des modèles intégrés.

Rubriques

- Opérations sur un seul document
- Opérations sur plusieurs documents
- Limitation des demandes pour les demandes en temps réel (synchrones)

Opérations sur un seul document

L'API Amazon Comprehend fournit des opérations qui prennent un seul document en entrée. Les quotas suivants s'appliquent à ces opérations.

Quotas généraux pour les opérations portant sur un seul document

Les quotas suivants s'appliquent à l'analyse en temps réel visant à détecter les entités, les phrases clés ou le langage dominant. Pour la détection des entités, ces quotas s'appliquent à la détection à l'aide des modèles intégrés. Pour la détection d'entités personnalisées, consultez les quotas dansReconnaissance d'entités personnalisée .

Description	Quot/ligne directrice
Taille maximale du document	100 Ko

Quotas spécifiques aux opérations pour les opérations portant sur un seul document

Les quotas suivants s'appliquent à l'analyse en temps réel pour détecter le sentiment, le sentiment ciblé et la syntaxe.

Description	Quot/ligne directrice
Taille maximale du document	5 KO

Opérations sur plusieurs documents

L'API Amazon Comprehend fournit des opérations par lots qui traitent plusieurs documents avec une seule demande d'API. Les quotas suivants s'appliquent aux opérations par lots.

Description	Quot/ligne directrice
Taille maximale du document	5 KO
Nombre maximum de documents par demande	25

Pour plus d'informations sur l'utilisation des opérations sur les documents par lots, consultezTraitement synchrone de plusieurs documents.

Limitation des demandes pour les demandes en temps réel (synchrones)

Amazon Comprehend applique une régulation dynamique aux demandes synchrones. Si la bande passante de traitement du système est disponible, Amazon Comprehend augmente progressivement le nombre de demandes traitées. Pour contrôler l'utilisation des opérations d'API synchrones par votre application, nous vous recommandons d'activer les alertes de facturation ou d'implémenter une limitation de débit dans votre application.

Analyse asynchrone

Cette section décrit les quotas liés à l'analyse asynchrone à l'aide des modèles intégrés.

Les opérations d'API asynchrones prennent chacune en charge un maximum de 10 tâches actives. Pour consulter les quotas pour chaque opération d'API, consultez le tableau des quotas de service dans les points de terminaison Amazon Comprehend et les quotas dans le manuel Amazon Web Services General Reference.

Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas.

Rubriques

- Quotas généraux pour les opérations asynchrones
- Quotas spécifiques à l'opération pour les tâches asynchrones
- Limitation des demandes pour les demandes asynchrones

Quotas généraux pour les opérations asynchrones

Vous pouvez exécuter des tâches d'analyse asynchrones à l'aide de la console ou de l'une des opérations de l'API`Start*`. Pour plus d'informations sur les circonstances dans lesquelles utiliser les opérations asynchrones, consultez. Traitement par lots asynchrone Les quotas suivants s'appliquent à la plupart des `Start*` opérations d'API pour les modèles intégrés. Pour les exceptions, voirQuotas spécifiques à l'opération pour les tâches asynchrones.

Description	Quot/ligne directrice
Taille maximale de chaque document dans les tâches qui détectent les entités, les phrases clés, les informations personnelles et les langues	1 Mo
Taille totale maximale de tous les fichiers d'une demande	5 Go
Taille totale minimale de tous les fichiers d'une demande	500 octets
Nombre maximum de fichiers, un document par fichier	1 000 000
Nombre total maximum de lignes, un document par ligne	1 000 000

Quotas spécifiques à l'opération pour les tâches asynchrones

Cette section décrit les quotas pour des opérations asynchrones spécifiques. Si aucun quota n'est spécifié dans les tableaux suivants, la valeur du quota général s'applique.

Rubriques

- Sentiment
- Sentiment ciblé
- Événements
- Modélisation des rubriques

Sentiment

Les jobs de sentiment asynchrones, que vous créez avec l'StartSentimentDetectionJobopération, sont soumis aux quotas suivants.

Description	Quot/ligne directrice
Taille maximale de chaque document d'entrée	5 KO

Sentiment ciblé

Les jobs de sentiment ciblés asynchrones, que vous créez avec l'StartTargetedSentimentDetectionJobopération, ont les quotas suivants.

Description	Quot/ligne directrice
Formats de document pris en charge	UTF-8
Taille maximale de chaque document d'une tâche	10 Ko
Taille maximale de tous les documents d'une tâche	300 MO
Nombre maximum de fichiers, un document par fichier	30 000
Nombre total maximum de lignes, un document par ligne (pour tous les fichiers d'une demande)	30 000

Événements

Les tâches de détection d'événements asynchrones, que vous créez avec l'StartEventsDetectionJobopération, ont les quotas suivants.

Description	Quotas
Encodage de caractères	UTF-8
Taille totale de tous les fichiers d'une tâche	50 Mo
Taille maximale de chaque document d'une tâche	10 Ko
Nombre maximum de fichiers, un document par fichier	5 000
Nombre total maximum de lignes, un document par ligne (pour tous les fichiers demandés)	5 000

Modélisation des rubriques

Les tâches de modélisation de sujets asynchrones, que vous créez avec l'StartTopicsDetectionJobopération, ont les quotas suivants.

Description	Quot/ligne directrice
Encodage de caractères	UTF-8
Nombre maximum de sujets à renvoyer	100
Taille de fichier maximale pour un fichier, un document par fichier	100 Mo

Pour plus d’informations, consultez Modélisation des rubriques.

Limitation des demandes pour les demandes asynchrones

Chaque opération d'API asynchrone prend en charge un nombre maximum de demandes par seconde (par région, par compte), ainsi qu'un maximum de 10 tâches actives. Pour consulter les

quotas pour chaque opération d'API, consultez le tableau des quotas de service dans les points de terminaison Amazon Comprehend et les quotas dans le manuel Amazon Web Services General Reference.

Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas.

Quotas pour les modèles personnalisés

Vous pouvez utiliser Amazon Comprehend pour créer vos propres modèles personnalisés pour une classification personnalisée et une reconnaissance d'entité personnalisée. Cette section fournit les directives et les quotas relatifs à la formation et à l'utilisation de modèles personnalisés. Pour plus d'informations sur les modèles personnalisés, consultezAmazon Comprehend personnalisé.

Rubriques

- Quotas généraux
- Quotas pour les terminaux
- Classification des documents
- Reconnaissance d'entités personnalisée

Quotas généraux

Amazon Comprehend définit des quotas de taille généraux pour chaque type de document d'entrée que vous pouvez analyser à l'aide de modèles personnalisés. Pour les quotas d'analyse en temps réel, voirTailles de document maximales pour une analyse en temps réel. Pour les quotas d'analyse asynchrones, consultez. Entrées pour une analyse personnalisée asynchrone

Chaque opération d'API asynchrone prend en charge un nombre maximum de demandes par seconde (par région, par compte), ainsi qu'un maximum de 10 tâches actives. Pour consulter les quotas pour chaque opération d'API, consultez le tableau des quotas de service dans les points de terminaison Amazon Comprehend et les quotas dans le manuel Amazon Web Services General Reference.

Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas.

Quotas pour les terminaux

Vous créez un point de terminaison pour exécuter une analyse en temps réel avec un modèle personnalisé. Pour plus d'informations sur les points de terminaison, consultezGestion des points de terminaison Amazon Comprehend.

Les quotas suivants s'appliquent aux points de terminaison. Pour plus d'informations sur la procédure à suivre pour demander une augmentation de quota, consultez AWS Service Quotas.

Description	Quot/ligne directrice
Nombre maximum de points de terminaison actifs par région pour chaque compte	20
Nombre maximum d'unités d'inférence par région pour chaque compte	200
Nombre maximal d'unités d'inférence par point de terminaison par région	50
Débit maximal par unité d'inférence (caractères)	100/seconde
Débit maximal par unité d'inférence (documents)	2/seconde

Classification des documents

Cette section décrit les directives et les quotas pour les opérations de classification de documents suivantes :

- Classifiez les tâches de formation que vous commencez dès l'CreateDocumentClassifieropération.
- Tâches de classification de documents asynchrones que vous commencez par l'StartDocumentClassificationJobopération.
- Demandes de classification de documents synchrones qui utilisent l'ClassifyDocumentopération.

Quotas généraux pour le classement des documents

Le tableau suivant décrit les quotas généraux liés à la formation des classificateurs personnalisés.

Description	Quot/ligne directrice
Longueur maximale du nom de classe	5 000 caractères
Nombre de classes (mode multi-classes)	2 à 1 000
Nombre de classes (mode multi-étiquettes)	2 à 100
Format des annotations	
Nombre minimum d'annotations par classe (mode multi-classes)	10
Nombre minimum d'annotations par classe (mode multi-étiquettes)	10
Nombre minimum d'annotations (mode multi-étiquettes)	50
Format de fichier CSV	
Nombre minimum de documents de formation par classe (mode multi-cours)	50
Nombre minimum de documents de formation par classe (mode multi-étiquettes)	10
Nombre minimal de documents de formation (mode multi-étiquettes)	50

Classification des documents en texte brut

Vous créez et entraînez un modèle en texte brut à l'aide de documents d'entrée en texte brut. Amazon Comprehend fournit des opérations asynchrones et en temps réel pour classer les documents en texte brut à l'aide d'un modèle en texte brut.

Entraînement

Le tableau suivant décrit les quotas liés à l'entraînement d'un classificateur personnalisé avec des documents en texte brut.

Description	Quot/ligne directrice
Taille totale de tous les fichiers relatifs à la tâche de formation	5 Go
Nombre maximum de fichiers manifestes augmentés pour la formation d'un classificateur personnalisé	5
Nombre maximum de noms d'attributs pour chaque fichier manifeste augmenté	5
Longueur maximale du nom d'attribut	63 personnages

Analyse en temps réel (synchrone)

Le tableau suivant décrit les quotas liés à la classification en temps réel des documents en texte brut.

Description	Quot/ligne directrice
Nombre maximum de documents par demande synchrone	1
Taille maximale du document texte (encodé en UTF-8)	10 Ko

Analyse asynchrone

Le tableau suivant décrit les quotas liés à la classification asynchrone des documents en texte brut.

Description	Quot/ligne directrice
Taille totale de tous les fichiers dans une tâche asynchrone	5 Go
Taille de fichier maximale pour un fichier, un document par fichier	10 Mo
Nombre maximum de fichiers, un document par fichier	1 000 000
Nombre total maximum de lignes, un document par ligne (pour tous les fichiers demandés)	1 000 000

Classification des documents semi-structurés

Cette section décrit les directives et les quotas pour la classification des documents semi-structurés. Pour classer des documents semi-structurés, utilisez un modèle de document natif que vous avez entraîné avec des documents d'entrée natifs.

Formation d'un modèle de document natif avec des documents semi-structurés

Le tableau suivant décrit les quotas liés à l'entraînement d'un classificateur personnalisé avec des documents semi-structurés, tels que des documents PDF, des documents Word et des fichiers image.

Description	Quot/ligne directrice
Nombre maximum de pages pour tous les documents	10 000
Taille maximale du fichier d'annotations (toutes tailles de fichiers CSV combinées)	5 Mo
Taille du corpus de documents (documents de formation et de test)	10 Go
Tailles de fichiers pour les fichiers de formation et de test	
Taille du fichier image (JPG, PNG, TIFF).	1 octet à 10 Mo. Fichiers TIFF : une page maximum.
Taille de page pour les documents PDF	1 octet—10 Mo
Taille de page pour les documents Word	1 octet—10 Mo
Taille JSON de sortie de l'API Amazon Textract	1 octet—1 Mo

Analyse en temps réel (synchrone)

Cette section décrit les quotas liés à la classification en temps réel des documents semi-structurés.

Le tableau suivant indique les tailles de fichier maximales pour les documents d'entrée. Pour tous les types de documents d'entrée, le maximum du fichier d'entrée est d'une page, avec un maximum de 10 000 caractères.

Type de fichier	Taille maximale (API)	Taille maximale (console)
Documents texte UTF-8	10 Ko	10 Ko
Documents PDF	10 Mo	5 Mo
Documents Word	10 Mo	5 Mo
Fichiers image	10 Mo	5 Mo
Taille de sortie de l'API Amazon Textract	1 Mo	N/A

Analyse asynchrone

Le tableau suivant décrit les quotas liés à la classification asynchrone des documents semi-structurés.

Description	Quot/ligne directrice
Nombre maximal de pages pour tous les documents d'entrée pour une tâche	25 000
Taille du corpus de documents	25 Go
Taille du fichier image (JPG, PNG ou TIFF)	1 octet à 10 Mo. Fichiers TIFF : une page maximum.
Taille de page pour les documents PDF	1 octet—10 Mo
Taille de page pour les documents Word	1 octet—10 Mo
Taille JSON de sortie de l'API Textract	1 octet—1 Mo.

Reconnaissance d'entités personnalisée

Cette section décrit les directives et les quotas pour les opérations suivantes relatives à la reconnaissance d'entités personnalisées :

- Les tâches de formation relatives à la reconnaissance d'entités ont débuté lors de l'CreateEntityRecognizeropération.
- Les tâches de reconnaissance d'entités asynchrones ont démarré avec l'StartEntitiesDetectionJobopération.
- Demandes de reconnaissance d'entités synchrones utilisant l'DetectEntitiesopération.

Reconnaissance d'entités personnalisée pour les documents en texte brut

Amazon Comprehend propose des opérations asynchrones et de synchronisation pour analyser des documents en texte brut à l'aide d'un outil de reconnaissance d'entités personnalisé.

Entraînement

Cette section décrit les quotas liés à la formation d'un outil de reconnaissance d'entités personnalisé pour analyser des documents en texte brut. Pour entraîner le modèle, vous pouvez fournir une liste d'entités ou un ensemble de documents texte annotés.

Le tableau suivant décrit les quotas liés à l'entraînement du modèle avec une liste d'entités.

Description	Quot/ligne directrice
Nombre d'entités par modèle	1 à 25
Taille du document (UTF-8)	1 à 5 000 octets
Nombre d'éléments dans la liste des entités	1 à 1 million
Durée de l'inscription individuelle (post-strip) dans la liste des inscrits	1 à 5 000
Taille du corpus de la liste d'entités (tous les documents en texte brut combinés)	5 KO —200 MB

Le tableau suivant décrit les quotas liés à l'entraînement du modèle avec des documents texte annotés.

Description	Quot/ligne directrice
Nombre d'entités par modèle/outil de reconnaissance d'entités personnalisé	1 à 25
Taille du document (UTF-8)	1 à 5 000 octets
Nombre de documents (voir Annotations en texte brut)	3 à 200 000
Taille du corpus du document (tous les documents en texte brut combinés)	5 KO - 200 MB
Nombre minimum d'annotations par entité	25

Analyse en temps réel (synchrone)

Le tableau suivant décrit les quotas liés à l'analyse en temps réel de documents en texte brut.

Description	Quot/ligne directrice
Nombre maximum de documents par demande synchrone	1
Taille maximale du document texte (encodé en UTF-8)	5 KO

Analyse asynchrone

Le tableau suivant décrit les quotas liés à la reconnaissance d'entités asynchrones de documents en texte brut.

Description	Quot/ligne directrice
Taille du document (UTF-8)	1 octet—1 Mo
Nombre maximum de fichiers, un document par fichier	1 000 000

Description	Quot/ligne directrice
Nombre total maximum de lignes, un document par ligne (pour tous les fichiers demandés)	1 000 000
Taille du corpus du document (tous les documents en texte brut combinés)	1 octet—5 Go

Reconnaissance d'entités personnalisée pour les documents semi-structurés

Amazon Comprehend propose des opérations asynchrones et de synchronisation pour analyser des documents semi-structurés à l'aide d'un outil de reconnaissance d'entités personnalisé. Vous devez entraîner le modèle à l'aide de documents PDF annotés.

Entraînement

Le tableau suivant décrit les quotas liés à la formation d'un outil de reconnaissance d'entités personnalisé (CreateEntityRecognizer) pour analyser des documents semi-structurés.

Description	Quot/ligne directrice
Nombre d'entités par modèle/outil de reconnaissance d'entités personnalisé	1 à 25
Taille maximale du fichier d'annotation (UTF-8 JSON)	5 Mo
Nombre de documents	250 à 10 000
Taille du corpus du document (tous les documents en texte brut combinés)	5 KO À 1 GO
Nombre minimum d'annotations par entité	100
Nombre maximum de fichiers manifestes augmentés pour la formation d'un outil de reconnaissance d'entités personnalisé	5
Nombre maximum de noms d'attributs pour chaque fichier manifeste augmenté	5

Description	Quot/ligne directrice
Longueur maximale du nom d'attribut	63 personnages

Analyse en temps réel (synchrone)

Cette section décrit les quotas liés à l'analyse en temps réel de documents semi-structurés.

Le tableau suivant indique les tailles de fichier maximales pour les documents d'entrée. Pour tous les types de documents d'entrée, le maximum du fichier d'entrée est d'une page, avec un maximum de 10 000 caractères.

Type de fichier	Taille maximale (API)	Taille maximale (console)
Documents texte UTF-8	10 Ko	10 Ko
Documents PDF	10 Mo	5 Mo
Documents Word	10 Mo	5 Mo
Fichiers image	10 Mo	5 Mo
Textract les fichiers de sortie	1 Mo	N/A

Analyse asynchrone

Cette section décrit les quotas pour l'analyse asynchrone de documents semi-structurés.

Description	Quot/ligne directrice
Taille de l'image (JPG ou PNG)	1 octet—10 Mo
Taille de l'image (TIFF)	1 octet à 10 Mo. Une page au maximum.
Taille du document (PDF)	1 octet—50 Mo
Taille du document (Docx)	1 octet—5 Mo

Description	Quot/ligne directrice
Taille du document (UTF-8)	1 octet—1 Mo
Nombre maximum de fichiers, un document par fichier (un document par ligne n'est pas autorisé pour les fichiers image ou les documents PDF/Word)	500
Nombre maximum de pages pour un fichier PDF ou Docx	100
Taille du corpus du document après extraction du texte (texte brut, tous fichiers confondus)	1 octet—5 Go

Pour plus d'informations sur les limites applicables aux images, consultez la section Limites strictes dans Amazon Textract

Quotas pour les volants

Utilisez des volants pour gérer la formation et le suivi des versions de modèles personnalisés à des fins de classification et de reconnaissance d'entités personnalisées. Pour plus d'informations sur Flywheels, voirVolants.

Quotas généraux pour les volants

Les quotas suivants s'appliquent aux volants et aux itérations de volants.

Description	Quot/ligne directrice
Nombre maximum de volants	50
Nombre maximum de volants en état CREATING	10
Nombre maximum de jeux de données d'entraînement par volant	50
Nombre maximum de jeux de données de test par volant	50

Description	Quot/ligne directrice
Nombre maximum de jeux de données ayant le statut INGESTING	10
Nombre maximum d'itérations de volant en cours par compte	10

Quotas de jeux de données pour les modèles de classification personnalisés

Lorsque vous ingérez un jeu de données pour un volant associé à un modèle de classification personnalisé, les quotas suivants s'appliquent.

Description	Quot/ligne directrice
Nombre minimum de documents de formation par classe (mode multi-étiquettes)	50
Nombre maximum de documents de formation	1 000 000
Taille minimale du jeu de données	500 octets
Taille maximale du jeu de données	5 Go
Taille de fichier maximale pour un fichier, un document par fichier	10 Mo

Quotas de jeux de données pour les modèles de reconnaissance d'entités personnalisés

Lorsque vous ingérez un ensemble de données pour un volant associé à un modèle de reconnaissance d'entité personnalisé, les quotas suivants s'appliquent.

Description	Quot/ligne directrice
Taille maximale du document	5 KO

Description	Quot/ligne directrice
Nombre minimum de documents de formation	3
Nombre maximum de documents de formation	200 000
Nombre minimum d'annotations par entité	25
Taille maximale du jeu de données	200 Mo

Tutoriels et autres ressources

Tutoriels et autres ressources pour Amazon Comprehend.

Rubriques

- Tutoriel : Analyse des informations issues des avis clients avec Amazon Comprehend
- Utilisation des points d'accès Lambda des objets Amazon S3 pour les informations personnelles identifiables (PII)
- Solution : analyser du texte avec Amazon Comprehend et OpenSearch

Tutoriel : Analyse des informations issues des avis clients avec Amazon Comprehend

Ce didacticiel explique comment utiliser Amazon Comprehend avec Amazon Simple Storage Service, AWS GlueAmazon Athena, et Amazon QuickSight pour obtenir des informations précieuses sur vos documents. Amazon Comprehend peut extraire des sentiments (l'ambiance d'un document) et des entités (noms de personnes, d'organisations, d'événements, de dates, de produits, de lieux, de quantités et de titres) à partir de texte non structuré.

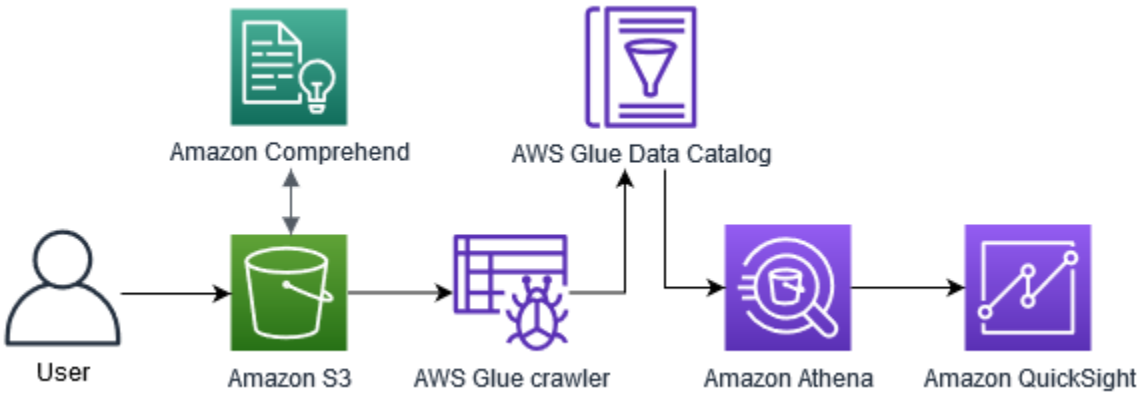
Par exemple, vous pouvez obtenir des informations exploitables à partir des avis des clients. Dans ce didacticiel, vous allez analyser un ensemble de données d'avis clients sur un roman. Vous utilisez l'analyse des sentiments d'Amazon Comprehend pour déterminer si les clients ont une opinion positive ou négative du roman. Vous utilisez également l'analyse des entités Amazon Comprehend pour découvrir les mentions d'entités importantes, telles que des romans ou des auteurs connexes. Après avoir suivi ce tutoriel, vous découvrirez peut-être que plus de 50 % des avis sont positifs. Vous pourriez également découvrir que les clients comparent les auteurs et manifestent de l'intérêt pour d'autres romans classiques.

Dans ce didacticiel, vous effectuez les opérations suivantes :

- Stockez un exemple de jeu de données d'avis dans Amazon Simple Storage Service (Amazon S3). Amazon Simple Storage Service est un service de stockage d'objets.
- Utilisez Amazon Comprehend pour analyser le sentiment et les entités contenus dans les documents de révision.

- Utilisez un AWS Glueobot pour stocker les résultats de l'analyse dans une base de données. AWS Glue est un service Extraction, transformation et chargement (ETL) qui vous permet de cataloguer et de nettoyer vos données à des fins d'analyse.
- Exécutez Amazon Athenades requêtes pour nettoyer vos données. Amazon Athena est un service de requêtes interactif sans serveur.
- Créez des visualisations avec vos données sur Amazon QuickSight. QuickSight est un outil de business intelligence sans serveur qui permet d'extraire des informations de vos données.

Le schéma suivant illustre le flux de travail.



Temps estimé pour terminer ce didacticiel : 1 heure

Coût estimé : Certaines des actions de ce didacticiel entraînent des frais pour votre Compte AWS. Pour de plus amples informations sur les frais de chacun de ces services, veuillez consulter les pages de tarification suivantes.

- Tarification d'Amazon S3
- Tarification d'Amazon Comprehend
- AWS Glue tarification
- Tarification Amazon Athena
- Tarification QuickSight

Rubriques

- Prérequis
- Étape 1 : ajout de documents à Amazon S3
- Étape 2 : (CLI uniquement) création d'un rôle IAM pour Amazon Comprehend

- Étape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3
- Étape 4 : Préparation de la sortie Amazon Comprehend pour la visualisation des données
- Étape 5 : Visualisation du résultat d'Amazon Comprehend dans QuickSight

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des éléments suivants :

- Un Compte AWS. Pour plus d'informations sur la configuration d'un Compte AWS, consultezConfiguration.
- Une IAM (utilisateur, groupe ou rôle). Pour savoir comment configurer des utilisateurs et des groupes pour votre compte, consultez le didacticiel de démarrage du guide de l'utilisateur IAM.
- La politique d'autorisation suivante est associée à votre utilisateur, groupe ou rôle. La politique accorde certaines autorisations requises pour terminer ce didacticiel. Le prérequis suivant décrit les autorisations supplémentaires dont vous avez besoin.

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action":
      [
        "comprehend:*",
        "ds:AuthorizeApplication",
        "ds:CheckAlias",
        "ds:CreateAlias",
        "ds:CreateIdentityPoolDirectory",
        "ds:DeleteDirectory",
        "ds:DescribeDirectories",
        "ds:DescribeTrusts",
        "ds:UnauthorizeApplication",
        "iam:AttachRolePolicy",
        "iam:CreatePolicy",
        "iam:CreatePolicyVersion",
        "iam:CreateRole",
        "iam:DeletePolicyVersion",
        "iam:DeleteRole",
```

```
        "iam:DetachRolePolicy",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:ListAccountAliases",
        "iam:ListAttachedRolePolicies",
        "iam:ListEntitiesForPolicy",
        "iam:ListPolicies",
        "iam:ListPolicyVersions",
        "iam:ListRoles",
        "quicksight:*",
        "s3:*",
        "tag:GetResources"
      ],
      "Resource": "*"
    },
    {
      "Action":
      [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:iam::*:role/*Comprehend*"
      ]
    }
  ]
}
```

Utilisez la politique précédente pour créer une stratégie IAM et l'associer à votre groupe ou utilisateur. Pour plus d'informations sur la création d'une stratégie IAM, consultez Création de stratégies IAM dans le IAM Guide de l'utilisateur. Pour plus d'informations sur l'ajout d'une politique IAM, consultez la rubrique Ajout et suppression d'autorisations basées sur l'identité IAM dans le Guide de l'utilisateur IAM.

- Politiques gérées attachées à votre groupe ou utilisateur IAM. Outre la politique précédente, vous devez également attacher les stratégies AWS gérées suivantes à votre groupe ou utilisateur :
  - `AWSGlueConsoleFullAccess`
  - `AWSQuicksightAthenaAccess`

Ces politiques gérées vous donnent l'autorisation d'utiliser AWS Glue Amazon Athena, et QuickSight. Pour plus d'informations sur l'ajout d'une politique IAM, consultez la rubrique Ajout et suppression d'autorisations basées sur l'identité IAM dans le Guide de l'utilisateur IAM.

Étape 1 : ajout de documents à Amazon S3

Avant de commencer les tâches d'analyse Amazon Comprehend, vous devez stocker un exemple de jeu de données d'avis clients dans Amazon Simple Storage Service (Amazon S3). Amazon S3 héberge vos données dans des conteneurs appelés buckets. Amazon Comprehend peut analyser les documents stockés dans un compartiment et envoyer les résultats de l'analyse à un compartiment. Au cours de cette étape, vous allez créer un compartiment S3, créer des dossiers d'entrée et de sortie dans le compartiment et télécharger un exemple de jeu de données dans le compartiment.

Rubriques

- Prérequis
- Télécharger des exemples de données
- Créer un compartiment Amazon S3
- (Console uniquement) créer des dossiers
- Téléchargez les données d'entrée

Prérequis

Avant de commencer, passez en revue Tutoriel : Analyse des informations issues des avis clients avec Amazon Comprehend et complétez les prérequis.

Télécharger des exemples de données

L'exemple de jeu de données suivant contient des critiques Amazon extraites du jeu de données plus vaste « Amazon reviews - Full », publié avec l'article « Character-level Convolutional Networks for Text Classification » (Xiang Zhang et al., 2015). Téléchargez le jeu de données sur votre ordinateur.

Pour obtenir les exemples de données

1. Téléchargez le fichier zip tutorial-reviews-data.zip sur votre ordinateur.
2. Décompressez le fichier zip sur votre ordinateur. Il y a deux fichiers. Le fichier `THIRD_PARTY_LICENSES.txt` est la licence open source pour le jeu de données publié par

Xiang Zhang et al. Le fichier `amazon-reviews.csv` est le jeu de données que vous analysez dans le didacticiel.

Créer un compartiment Amazon S3

Après avoir téléchargé l'exemple de jeu de données, créez un compartiment Amazon S3 pour stocker vos données d'entrée et de sortie. Vous pouvez créer un compartiment S3 à l'aide de la console Amazon S3 ou du AWS Command Line Interface (AWS CLI).

Création d'un compartiment Amazon S3 (console)

Dans la console Amazon S3, vous créez un compartiment dont le nom est unique dans tous les compartiments AWS.

Pour créer un compartiment S3 (console)

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans Buckets, choisissez Create bucket.
3. Dans le champ Nom du compartiment, entrez un nom unique au monde qui décrit l'objectif du compartiment.
4. Pour Région, choisissez la AWS région dans laquelle vous souhaitez créer le bucket. La région que vous choisissez doit prendre en charge Amazon Comprehend. Pour réduire le temps de latence, choisissez la AWS région la plus proche de votre situation géographique prise en charge par Amazon Comprehend. Pour obtenir la liste des régions qui prennent en charge Amazon Comprehend, consultez le tableau des régions du Guide mondial de l'infrastructure.
5. Conservez les paramètres par défaut pour la propriété de l'objet, les paramètres du compartiment pour le blocage de l'accès public, le versionnement des compartiments et les balises.
6. Pour le chiffrement par défaut, choisissez Désactiver.

> **Tip**
>
> Bien que ce didacticiel n'utilise pas le chiffrement, vous souhaiterez peut-être utiliser le chiffrement lors de l'analyse de données importantes. Pour end-to-end le chiffrement, vous pouvez chiffrer vos données au repos dans le compartiment et également lorsque vous exécutez des tâches d'analyse. Pour plus d'informations sur le chiffrement avec

AWS, voir Qu'est-ce que c'est AWS Key Management Service ? dans le Guide AWS Key Management Service du développeur.

7. Passez en revue les configurations de votre compartiment, puis choisissez Create bucket.

Création d'un compartiment Amazon S3 (AWS CLI)

Après avoir ouvert le AWS CLI, vous exécutez la `create-bucket` commande pour créer un compartiment qui stockera les données d'entrée et de sortie.

Pour créer un compartiment Amazon S3 (AWS CLI)

1. Pour créer votre bucket, exécutez la commande suivante dans le AWS CLI. Remplacez amzn-s3-demo-bucket par un nom unique pour le bucket. AWS

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket
```

Par défaut, la `create-bucket` commande crée un compartiment dans la `us-east-1` AWS région. Pour créer un compartiment dans un Région AWS autre que`us-east-1`, ajoutez le `LocationConstraint` paramètre pour spécifier votre région. Par exemple, la commande suivante crée un compartiment dans la `us-west-2` région.

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket
--region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2
```

Notez que seules certaines régions prennent en charge Amazon Comprehend. Pour obtenir la liste des régions qui prennent en charge Amazon Comprehend, consultez le tableau des régions du Guide mondial de l'infrastructure.

2. Pour vous assurer que votre bucket a bien été créé, exécutez la commande suivante. La commande répertorie tous les compartiments S3 associés à votre compte.

```
aws s3 ls
```

(Console uniquement) créer des dossiers

Créez ensuite deux dossiers dans votre compartiment S3. Le premier dossier est destiné à vos données d'entrée. Le deuxième dossier est l'endroit où Amazon Comprehend envoie les résultats

de l'analyse. Si vous utilisez la console Amazon S3, vous devez créer les dossiers manuellement. Si vous utilisez le AWS CLI, vous pouvez créer des dossiers lorsque vous chargez l'exemple de jeu de données ou que vous exécutez une tâche d'analyse. C'est pourquoi nous proposons une procédure permettant de créer des dossiers uniquement pour les utilisateurs de la console. Si vous utilisez la AWS CLI, vous allez créer des dossiers dans Téléchargez les données d'entrée et dansÉtape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3.

Pour créer des dossiers dans votre compartiment S3 (console)

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans Buckets, choisissez votre bucket dans la liste des buckets.
3. Dans l'onglet Vue d'ensemble, choisissez Créer un dossier.
4. Pour le nouveau nom du dossier, entrez`input`.
5. Pour les paramètres de chiffrement, choisissez Aucun (Utiliser les paramètres du bucket).
6. Choisissez Save (Enregistrer).
7. Répétez les étapes 3 à 6 pour créer un autre dossier pour la sortie des tâches d'analyse, mais à l'étape 4, entrez le nouveau nom du dossier`output`.

Téléchargez les données d'entrée

Maintenant que vous avez un bucket, chargez l'exemple de jeu de donnéesamazon-reviews.csv. Vous pouvez télécharger des données dans des compartiments S3 à l'aide de la console Amazon S3 ou du AWS CLI.

Charger des exemples de documents dans un bucket (console)

Dans la console Amazon S3, chargez le fichier d'exemple de jeu de données dans le dossier d'entrée.

Pour télécharger les exemples de documents (console)

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans Buckets, choisissez votre bucket dans la liste des buckets.
3. Choisissez le `input` dossier, puis choisissez Upload.
4. Choisissez Ajouter des fichiers, puis sélectionnez le `amazon-reviews.csv` fichier sur votre ordinateur.
5. Conservez les valeurs par défaut des autres paramètres.

6. Choisissez Charger.

Charger des exemples de documents dans un bucket (AWS CLI)

Créez un dossier d'entrée dans votre compartiment S3 et téléchargez le fichier du jeu de données dans le nouveau dossier à l'aide de la `cp` commande.

Pour télécharger les exemples de documents (AWS CLI)

1. Pour télécharger le `amazon-reviews.csv` fichier dans un nouveau dossier de votre bucket, exécutez la AWS CLI commande suivante. Remplacez amzn-s3-demo-bucket par le nom de votre bucket. En ajoutant le chemin `/input/` à la fin, Amazon S3 crée automatiquement un nouveau dossier appelé `input` dans votre compartiment et télécharge le fichier du jeu de données dans ce dossier.

   ```
   aws s3 cp amazon-reviews.csv s3://amzn-s3-demo-bucket/input/
   ```

2. Pour vous assurer que votre fichier a bien été chargé, exécutez la commande suivante. La commande répertorie le contenu du `input` dossier de votre bucket.

   ```
   aws s3 ls s3://amzn-s3-demo-bucket/input/
   ```

Vous disposez maintenant d'un compartiment S3 contenant le `amazon-reviews.csv` fichier dans un dossier appelé`input`. Si vous avez utilisé la console, le compartiment contient également un `output` dossier. Si vous avez utilisé le AWS CLI, vous créerez le dossier de sortie lors de l'exécution des tâches d'analyse Amazon Comprehend.

Étape 2 : (CLI uniquement) création d'un rôle IAM pour Amazon Comprehend

Cette étape n'est nécessaire que si vous utilisez le AWS Command Line Interface (AWS CLI) pour terminer ce didacticiel. Si vous utilisez la console Amazon Comprehend pour exécuter les tâches d'analyse, passez directement à. Étape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3

Pour exécuter des tâches d'analyse, Amazon Comprehend doit accéder au compartiment Amazon S3 qui contient l'exemple de jeu de données et contiendra les résultats des tâches. Les rôles IAM vous permettent de contrôler les autorisations des AWS services ou des utilisateurs. Au cours de cette

étape, vous allez créer un rôle IAM pour Amazon Comprehend. Ensuite, vous créez et associez à ce rôle une politique basée sur les ressources qui accorde à Amazon Comprehend l'accès à votre compartiment S3. À la fin de cette étape, Amazon Comprehend aura les autorisations nécessaires pour accéder à vos données d'entrée, stocker vos résultats et exécuter des tâches d'analyse des sentiments et des entités.

Pour plus d'informations sur l'utilisation d'IAM avec Amazon Comprehend, consultez. Comment Amazon Comprehend fonctionne avec IAM

Rubriques

- Prérequis
- Créer un rôle IAM
- Associer une politique IAM au rôle IAM

Prérequis

Avant de commencer, vous devez exécuter les actions suivantes :

- Termin Étape 1 : ajout de documents à Amazon S3.
- Utilisez un éditeur de code ou de texte pour enregistrer les politiques JSON et suivre vos Amazon Resource Names (ARNs).

Créer un rôle IAM

Pour accéder à votre compartiment Amazon Simple Storage Service (Amazon S3), Amazon Comprehend doit assumer AWS Identity and Access Management un rôle (IAM). Le rôle IAM déclare Amazon Comprehend en tant qu'entité de confiance. Une fois qu'Amazon Comprehend a assumé le rôle et est devenu une entité de confiance, vous pouvez accorder des autorisations d'accès au bucket à Amazon Comprehend. Au cours de cette étape, vous créez un rôle qui identifie Amazon Comprehend comme une entité de confiance. Vous pouvez créer un rôle à l'aide de la console Amazon Comprehend AWS CLI ou de la console Amazon Comprehend. Pour utiliser la console, passez àÉtape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3.

La console Amazon Comprehend vous permet de sélectionner des rôles dont le nom contient « Comprehend » et dont la politique de confiance inclut le nom. comprehend.amazonaws.com Configurez les rôles créés par la CLI pour répondre à ces critères si vous souhaitez que la console les affiche.

Pour créer un rôle IAM pour Amazon Comprehend (AWS CLI)

1. Enregistrez la politique de confiance suivante sous forme de document JSON appelé `comprehend-trust-policy.json` dans un éditeur de code ou de texte sur votre ordinateur. Cette politique de confiance déclare Amazon Comprehend comme une entité de confiance et lui permet d'assumer un rôle IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Pour créer le rôle IAM, exécutez la AWS CLI commande suivante. La commande crée un rôle IAM appelé `AmazonComprehendServiceRole-access-role` et associe la politique de confiance au rôle. *path/*Remplacez-le par le chemin de votre ordinateur local vers le document JSON.

```
aws iam create-role --role-name AmazonComprehendServiceRole-access-role
--assume-role-policy-document file://path/comprehend-trust-policy.json
```

> **Tip**
>
> Si vous recevez un message d'erreur lors de l'analyse des paramètres, le chemin d'accès à votre fichier de politique de confiance JSON est probablement incorrect. Indiquez le chemin relatif du fichier en fonction de votre répertoire personnel.

3. Copiez le nom de ressource Amazon (ARN) et enregistrez-le dans un éditeur de texte. Le format de l'ARN est similaire à*arn:aws:iam::123456789012:role/AmazonComprehendServiceRole-access-role*. Vous avez besoin de cet ARN pour exécuter les tâches d'analyse Amazon Comprehend.

Associer une politique IAM au rôle IAM

Pour accéder à votre compartiment Amazon S3, Amazon Comprehend a besoin d'autorisations pour répertorier, lire et écrire. Pour accorder à Amazon Comprehend les autorisations requises, créez et associez une politique IAM à votre rôle IAM. La politique IAM permet à Amazon Comprehend de récupérer les données d'entrée de votre compartiment et d'y écrire les résultats d'analyse. Après avoir créé la politique, vous l'associez à votre rôle IAM.

Pour créer une politique IAM ()AWS CLI

1. Enregistrez la politique suivante localement sous la forme d'un document JSON appelé`comprehend-access-policy.json`. Il accorde à Amazon Comprehend l'accès au compartiment S3 spécifié.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ],
```

```
            "Effect": "Allow"
        }
    ]
}
```

2. Pour créer la politique d'accès au compartiment S3, exécutez la AWS CLI commande suivante. *path/*Remplacez-le par le chemin de votre ordinateur local vers le document JSON.

```
aws iam create-policy --policy-name comprehend-access-policy
--policy-document file://path/comprehend-access-policy.json
```

3. Copiez l'ARN de la politique d'accès et enregistrez-le dans un éditeur de texte. Le format de l'ARN est similaire à*arn:aws:iam::123456789012:policy/comprehend-access-policy*. Vous avez besoin de cet ARN pour associer votre politique d'accès à votre rôle IAM.

Pour associer la politique IAM à votre rôle IAM ()AWS CLI

- Exécutez la commande suivante. *policy-arn*Remplacez-le par l'ARN de la politique d'accès que vous avez copié à l'étape précédente.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name AmazonComprehendServiceRole-access-role
```

Vous disposez désormais d'un rôle IAM doté d'une politique de confiance pour Amazon Comprehend et d'une politique d'accès qui accorde à Amazon Comprehend l'accès à votre compartiment S3. `AmazonComprehendServiceRole-access-role` L'ARN du rôle IAM est également copié dans un éditeur de texte.

Étape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3

Après avoir stocké les données dans Amazon S3, vous pouvez commencer à exécuter les tâches d'analyse Amazon Comprehend. Un travail d'analyse des sentiments détermine l'ambiance générale d'un document (positive, négative, neutre ou mixte). Une tâche d'analyse d'entités extrait les noms d'objets du monde réel d'un document. Ces objets incluent des personnes, des lieux, des titres, des événements, des dates, des quantités, des produits et des organisations. Au cours de cette étape, vous devez exécuter deux tâches d'analyse Amazon Comprehend pour extraire le sentiment et les entités de l'exemple de jeu de données.

Rubriques

- Prérequis
- Analyser les sentiments et les entités

Prérequis

Avant de commencer, vous devez exécuter les actions suivantes :

- Termin Étape 1 : ajout de documents à Amazon S3.
- (Facultatif) Si vous utilisez le AWS CLI, complétez Étape 2 : (CLI uniquement) création d'un rôle IAM pour Amazon Comprehend et préparez l'ARN de votre rôle IAM.

Analyser les sentiments et les entités

La première tâche que vous exécutez analyse le sentiment exprimé par chaque avis client dans l'exemple de jeu de données. La deuxième tâche extrait les entités de chaque avis client. Vous pouvez effectuer des tâches d'analyse Amazon Comprehend à l'aide de la console Amazon Comprehend ou du. AWS CLI

> **Tip**
>
> Assurez-vous que vous vous trouvez dans une AWS région qui prend en charge Amazon Comprehend. Pour plus d'informations, consultez le tableau des régions dans le Guide mondial de l'infrastructure.

Analyser les sentiments et les entités (console)

Lorsque vous utilisez la console Amazon Comprehend, vous créez une tâche à la fois. Vous devez répéter les étapes suivantes afin d'exécuter à la fois une tâche d'analyse des sentiments et une tâche d'analyse des entités. Notez que pour le premier travail, vous créez un rôle IAM, mais pour le second, vous pouvez réutiliser le rôle IAM du premier travail. Vous pouvez réutiliser le rôle IAM tant que vous utilisez le même compartiment et les mêmes dossiers S3.

Pour exécuter des tâches d'analyse des sentiments et des entités (console)

1. Assurez-vous que vous vous trouvez dans la même région que celle dans laquelle vous avez créé votre bucket Amazon Simple Storage Service (Amazon S3). Si vous vous trouvez dans une

autre région, dans la barre de navigation, choisissez la AWS région dans laquelle vous avez créé votre compartiment S3 dans le sélecteur de région.

2. Ouvrez la console Amazon Comprehend à l'adresse https://console.aws.amazon.com/comprehend/
3. Choisissez Launch Amazon Comprehend.
4. Dans le volet de navigation, sélectionnez Analysis jobs.
5. Choisissez Créer une tâche.
6. Dans la section Paramètres du job, procédez comme suit :
   a. Pour Nom, saisissez `reviews-sentiment-analysis`.
   b. Pour le type d'analyse, choisissez Sentiment.
   c. Dans Langue, choisissez l'anglais.
   d. Laissez le paramètre de chiffrement Job désactivé.
7. Dans la section Données d'entrée, procédez comme suit :
   a. Pour Source de données, sélectionnez Mes documents.
   b. Pour l'emplacement S3, choisissez Browse S3, puis choisissez votre bucket dans la liste des buckets.
   c. Dans votre compartiment S3, pour Objects, choisissez votre `input` dossier.
   d. Dans le `input` dossier, choisissez l'exemple de jeu de données, `amazon-reviews.csv` puis choisissez Choisir.
   e. Pour Format d'entrée, choisissez Un document par ligne.
8. Dans la section Données de sortie, procédez comme suit :
   a. Pour l'emplacement S3, choisissez Browse S3, puis choisissez votre bucket dans la liste des buckets.
   b. Dans votre compartiment S3, pour Objects, choisissez le `output` dossier, puis choisissez Choose.
   c. Laissez le chiffrement désactivé.
9. Dans la section Autorisations d'accès, procédez comme suit :
   a. Pour le rôle IAM, choisissez Create an IAM role.
   b. Pour les autorisations d'accès, choisissez les compartiments S3 d'entrée et de sortie.

c. Dans le champ Suffixe du nom, entrez`comprehend-access-role`. Ce rôle permet d'accéder à votre compartiment Amazon S3.

10. Choisissez Créer une tâche.
11. Répétez les étapes 1 à 10 pour créer une tâche d'analyse d'entités. Effectuez les modifications suivantes :

    a. Dans Paramètres du Job, pour Nom, entrez`reviews-entities-analysis`.

    b. Dans les paramètres du Job, pour Type d'analyse, sélectionnez Entities.

    c. Dans Autorisations d'accès, choisissez Utiliser un rôle IAM existant. Dans Nom du rôle, choisissez `AmazonComprehendServiceRole-comprehend-access-role` (il s'agit du même rôle que celui que vous avez créé pour le job d'opinion).

Analyser les sentiments et les entités (AWS CLI)

Vous utilisez les `start-entities-detection-job` commandes `start-sentiment-detection-job` et pour exécuter des tâches d'analyse des sentiments et des entités. Après avoir exécuté chaque commande, AWS CLI affiche un objet JSON avec une `JobId` valeur qui vous permet d'accéder aux détails de la tâche, y compris l'emplacement S3 de sortie.

Pour exécuter des tâches d'analyse des sentiments et des entités (AWS CLI)

1. Démarrez une tâche d'analyse des sentiments en exécutant la commande suivante dans le AWS CLI. *`arn:aws:iam::123456789012:role/comprehend-access-role`*Remplacez-le par l'ARN du rôle IAM que vous avez précédemment copié dans un éditeur de texte. Si votre AWS CLI région par défaut est différente de la région dans laquelle vous avez créé votre compartiment Amazon S3, incluez le `--region` paramètre et remplacez-le *`us-east-1`* par la région dans laquelle réside votre compartiment.

```
aws comprehend start-sentiment-detection-job
--input-data-config S3Uri=s3://amzn-s3-demo-bucket/input/
--output-data-config S3Uri=s3://amzn-s3-demo-bucket/output/
--data-access-role-arn arn:aws:iam::123456789012:role/comprehend-access-role
--job-name reviews-sentiment-analysis
--language-code en
[--region us-east-1]
```

2. Après avoir soumis le travail, copiez-le `JobId` et enregistrez-le dans un éditeur de texte. Vous en aurez besoin `JobId` pour rechercher les fichiers de sortie de la tâche d'analyse.

3. Démarrez une tâche d'analyse d'entités en exécutant la commande suivante.

```
aws comprehend start-entities-detection-job
--input-data-config S3Uri=s3://amzn-s3-demo-bucket/input/
--output-data-config S3Uri=s3://amzn-s3-demo-bucket/output/
--data-access-role-arn arn:aws:iam::123456789012:role/comprehend-access-role
--job-name reviews-entities-analysis
--language-code en
[--region us-east-1]
```

4. Après avoir soumis le travail, copiez-le `JobId` et enregistrez-le dans un éditeur de texte.
5. Vérifiez le statut de vos offres d'emploi. Vous pouvez suivre la progression d'une tâche en la suivant`JobId`.

   Pour suivre la progression de votre tâche d'analyse des sentiments, exécutez la commande suivante. *`sentiment-job-id`*Remplacez-le par celui `JobId` que vous avez copié après avoir effectué votre analyse des sentiments.

```
aws comprehend describe-sentiment-detection-job
--job-id sentiment-job-id
```

   Pour suivre votre tâche d'analyse des entités, exécutez la commande suivante. *`entities-job-id`*Remplacez-le par celui `JobId` que vous avez copié après avoir effectué votre analyse des entités.

```
aws comprehend describe-entities-detection-job
--job-id entities-job-id
```

   Il faut plusieurs minutes pour que le `JobStatus` s'affiche sous la forme`COMPLETED`.

Vous avez terminé des tâches d'analyse des sentiments et des entités. Les deux tâches doivent être terminées avant de passer à l'étape suivante. La fin des tâches peut prendre plusieurs minutes.

Étape 4 : Préparation de la sortie Amazon Comprehend pour la visualisation des données

Pour préparer les résultats des tâches d'analyse des sentiments et des entités afin de créer des visualisations de données, vous utilisez AWS Glue et Amazon Athena. Au cours de cette étape, vous

allez extraire les fichiers de résultats d'Amazon Comprehend. Ensuite, vous créez un AWS Glue robot d'exploration qui explore vos données et les catalogue automatiquement dans les tableaux du. AWS Glue Data Catalog Ensuite, vous pouvez accéder à ces tables et les transformer à l'aide Amazon Athena d'un service de requête interactif et sans serveur. Lorsque vous avez terminé cette étape, vos résultats Amazon Comprehend sont propres et prêts à être visualisés.

Pour une tâche de détection d'entités PII, le fichier de sortie est en texte brut et non une archive compressée. Le nom du fichier de sortie est le même que celui du fichier d'entrée, avec `.out` un ajout à la fin. Vous n'avez pas besoin de l'étape d'extraction du fichier de sortie. Passez à Charger les données dans un AWS Glue Data Catalog.

Rubriques

- Prérequis
- Téléchargez le résultat
- Extraire les fichiers de sortie
- Téléchargez les fichiers extraits
- Chargez les données dans un AWS Glue Data Catalog
- Préparer les données pour l'analyse

Prérequis

Avant de commencer, complétez Étape 3 : Exécution de tâches d'analyse sur des documents dans Amazon S3.

Téléchargez le résultat

Amazon Comprehend utilise la compression Gzip pour compresser les fichiers de sortie et les enregistrer sous forme d'archive tar. Le moyen le plus simple d'extraire les fichiers de sortie est de télécharger les `output.tar.gz` archives localement.

Au cours de cette étape, vous allez télécharger les archives de sortie des sentiments et des entités.

Télécharger les fichiers de sortie (console)

Pour trouver les fichiers de sortie pour chaque tâche, revenez à la tâche d'analyse dans la console Amazon Comprehend. La tâche d'analyse fournit l'emplacement S3 de la sortie, où vous pouvez télécharger le fichier de sortie.

Pour télécharger les fichiers de sortie (console)

1. Dans la console Amazon Comprehend, dans le volet de navigation, revenez à Analysis jobs.
2. Choisissez votre job d'analyse des sentimentsreviews-sentiment-analysis.
3. Sous Sortie, choisissez le lien affiché à côté de Emplacement des données de sortie. Cela vous redirige vers l'output.tar.gzarchive de votre compartiment S3.
4. Dans l'onglet Vue d'ensemble, choisissez Télécharger.
5. Sur votre ordinateur, renommez l'archive ensentiment-output.tar.gz. Comme tous les fichiers de sortie portent le même nom, cela vous permet de suivre les fichiers de sentiments et d'entités.
6. Répétez les étapes 1 à 4 pour rechercher et télécharger le résultat de votre reviews-entities-analysis travail. Sur votre ordinateur, renommez l'archive enentities-output.tar.gz.

Téléchargez les fichiers de sortie (AWS CLI)

Pour rechercher les fichiers de sortie pour chaque tâche, utilisez le fichier JobId de la tâche d'analyse pour trouver l'emplacement S3 de la sortie. Utilisez ensuite la cp commande pour télécharger le fichier de sortie sur votre ordinateur.

Pour télécharger les fichiers de sortie (AWS CLI)

1. Pour répertorier les détails de votre tâche d'analyse des sentiments, exécutez la commande suivante. Remplacez *sentiment-job-id* par le sentiment JobId que vous avez enregistré.

   ```
   aws comprehend describe-sentiment-detection-job --job-id sentiment-job-id
   ```

   Si vous avez perdu la trace de votre offre d'emploiJobId, vous pouvez exécuter la commande suivante pour répertorier toutes les offres d'emploi qui vous intéressent et filtrer par nom votre offre d'emploi.

   ```
   aws comprehend list-sentiment-detection-jobs
   --filter JobName="reviews-sentiment-analysis"
   ```

2. Dans l'OutputDataConfigobjet, trouvez la S3Uri valeur. La S3Uri valeur doit être similaire au format suivant :*s3://amzn-s3-demo-bucket/.../output/output.tar.gz*. Copiez cette valeur dans un éditeur de texte.

3. Pour télécharger l'archive de sortie des sentiments dans votre répertoire local, exécutez la commande suivante. Remplacez le chemin du compartiment S3 par celui `S3Uri` que vous avez copié à l'étape précédente. *path/*Remplacez-le par le chemin du dossier vers votre répertoire local. Le nom `sentiment-output.tar.gz` remplace le nom de l'archive d'origine pour vous aider à suivre les fichiers de sentiments et d'entités.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz
path/sentiment-output.tar.gz
```

4. Pour répertorier les détails de votre tâche d'analyse des entités, exécutez la commande suivante.

```
aws comprehend describe-entities-detection-job
--job-id entities-job-id
```

   Si vous ne connaissez pas votre tâche`JobId`, exécutez la commande suivante pour répertorier toutes les tâches de vos entités et filtrer votre tâche par nom.

```
aws comprehend list-entities-detection-jobs
--filter JobName="reviews-entities-analysis"
```

5. À partir de l'`OutputDataConfig`objet figurant dans la description de poste de votre entité, copiez la `S3Uri` valeur.
6. Pour télécharger l'archive de sortie des entités dans votre répertoire local, exécutez la commande suivante. Remplacez le chemin du compartiment S3 par celui `S3Uri` que vous avez copié à l'étape précédente. *path/*Remplacez-le par le chemin du dossier vers votre répertoire local. Le nom `entities-output.tar.gz` remplace le nom de l'archive d'origine.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz
path/entities-output.tar.gz
```

Extraire les fichiers de sortie

Avant de pouvoir accéder aux résultats d'Amazon Comprehend, décompressez les archives des sentiments et des entités. Vous pouvez utiliser votre système de fichiers local ou un terminal pour décompresser les archives.

Extraire les fichiers de sortie (système de fichiers GUI)

Si vous utilisez macOS, double-cliquez sur l'archive dans le système de fichiers de votre interface graphique pour extraire le fichier de sortie de l'archive.

Si vous utilisez Windows, vous pouvez utiliser un outil tiers tel que 7-Zip pour extraire les fichiers de sortie dans votre système de fichiers GUI. Sous Windows, vous devez effectuer deux étapes pour accéder au fichier de sortie dans l'archive. Décompressez d'abord l'archive, puis extrayez-la.

Renommez le fichier de sentiment comme `sentiment-output` et le fichier d'entités comme `entities-output` pour faire la distinction entre les fichiers de sortie.

Extraire les fichiers de sortie (terminal)

Si vous utilisez Linux ou macOS, vous pouvez utiliser votre terminal standard. Si vous utilisez Windows, vous devez avoir accès à un environnement de type Unix, tel que Cygwin, pour exécuter des commandes tar.

Pour extraire le fichier de sortie des sentiments de l'archive des sentiments, exécutez la commande suivante sur votre terminal local.

```
tar -xvf sentiment-output.tar.gz --transform 's,^,sentiment-,'
```

Notez que le `--transform` paramètre ajoute le préfixe `sentiment-` au fichier de sortie à l'intérieur de l'archive, renommant le fichier en. `sentiment-output` Cela vous permet de faire la distinction entre les fichiers de sortie de sentiment et les fichiers de sortie des entités et d'éviter toute réécriture.

Pour extraire le fichier de sortie des entités de l'archive des entités, exécutez la commande suivante dans votre terminal local.

```
tar -xvf entities-output.tar.gz --transform 's,^,entities-,'
```

Le `--transform` paramètre ajoute le préfixe `entities-` au nom du fichier de sortie.

> **Tip**
>
> Pour réduire les coûts de stockage dans Amazon S3, vous pouvez compresser à nouveau les fichiers avec Gzip avant de les télécharger. Il est important de décompresser et de décompresser les archives d'origine, car elles ne AWS Glue peuvent pas lire

automatiquement les données d'une archive tar. Cependant, il AWS Glue peut lire des fichiers au format Gzip.

Téléchargez les fichiers extraits

Après avoir extrait les fichiers, chargez-les dans votre bucket. Vous devez stocker les fichiers de sortie des sentiments et des entités dans des dossiers séparés AWS Glue afin de lire correctement les données. Dans votre compartiment, créez un dossier pour les résultats de sentiment extraits et un second dossier pour les résultats des entités extraits. Vous pouvez créer des dossiers à l'aide de la console Amazon S3 ou du AWS CLI.

Chargez les fichiers extraits sur Amazon S3 (console)

Dans votre compartiment S3, créez un dossier pour le fichier de résultats de sentiment extrait et un dossier pour le fichier de résultats d'entités. Téléchargez ensuite les fichiers de résultats extraits dans leurs dossiers respectifs.

Pour télécharger les fichiers extraits sur Amazon S3 (console)

1. Ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans Buckets, choisissez votre bucket, puis choisissez Create folder.
3. Pour le nouveau nom du dossier, entrez `sentiment-results` et choisissez Enregistrer. Ce dossier contiendra le fichier de sortie des sentiments extrait.
4. Dans l'onglet Vue d'ensemble de votre bucket, dans la liste des contenus du bucket, choisissez le nouveau dossier`sentiment-results`. Choisissez Charger.
5. Choisissez Ajouter des fichiers, sélectionnez le `sentiment-output` fichier sur votre ordinateur local, puis cliquez sur Suivant.
6. Conservez les options Gérer les utilisateurs, Accès pour les autres Compte AWS et Gérer les autorisations publiques comme options par défaut. Choisissez Suivant.
7. Pour Classe de stockage, choisissez Standard. Conservez les options de chiffrement, de métadonnées et de balise par défaut. Choisissez Suivant.
8. Passez en revue les options de téléchargement, puis choisissez Upload.
9. Répétez les étapes 1 à 8 pour créer un dossier appelé `entities-results` et téléchargez-y le `entities-output` fichier.

Chargez les fichiers extraits sur Amazon S3 (AWS CLI)

Vous pouvez créer un dossier dans votre compartiment S3 lors du chargement d'un fichier à l'aide de la `cp` commande.

Pour télécharger les fichiers extraits sur Amazon S3 (AWS CLI)

1. Créez un dossier de sentiments et téléchargez-y votre fichier de sentiments en exécutant la commande suivante. *`path/`*Remplacez-le par le chemin local vers votre fichier de sortie de sentiment extrait.

   ```
   aws s3 cp path/sentiment-output s3://amzn-s3-demo-bucket/sentiment-results/
   ```

2. Créez un dossier de sortie d'entités et téléchargez-y votre fichier d'entités en exécutant la commande suivante. *`path/`*Remplacez-le par le chemin local vers le fichier de sortie de vos entités extraites.

   ```
   aws s3 cp path/entities-output s3://amzn-s3-demo-bucket/entities-results/
   ```

Chargez les données dans un AWS Glue Data Catalog

Pour enregistrer les résultats dans une base de données, vous pouvez utiliser un AWS Glue robot d'exploration. Un AWS Glue robot d'exploration analyse les fichiers et découvre le schéma des données. Il organise ensuite les données dans des tables dans une AWS Glue Data Catalog (base de données sans serveur). Vous pouvez créer un robot d'exploration à l'aide de la AWS Glue console ou du AWS CLI.

Chargez les données dans une AWS Glue Data Catalog (console)

Créez un AWS Glue robot d'exploration qui analyse vos `entities-results` dossiers `sentiment-results` et vos dossiers séparément. Un nouveau rôle IAM pour AWS Glue donne à l'explorateur l'autorisation d'accéder à votre compartiment S3. Vous créez ce rôle IAM lors de la configuration du robot d'exploration.

Pour charger les données dans une AWS Glue Data Catalog (console)

1. Assurez-vous que vous vous trouvez dans une région qui prend en charge AWS Glue. Si vous vous trouvez dans une autre région, dans la barre de navigation, choisissez une région prise en

charge dans le sélecteur de région. Pour obtenir la liste des régions qui prennent en charge les infrastructures AWS Glue, consultez le tableau des régions du Guide mondial des infrastructures.

2. Ouvrez la AWS Glue console à l'adresse https://console.aws.amazon.com/glue/.
3. Dans le volet de navigation, choisissez Crawlers, puis choisissez Ajouter un robot d'exploration.
4. Pour le nom du robot d'exploration, entrez `comprehend-analysis-crawler` puis choisissez Next.
5. Pour le type de source Crawler, choisissez Data stores, puis Next.
6. Pour Ajouter un magasin de données, procédez comme suit :

   a. Dans Choisir un magasin de données, choisissez S3.
   b. Laissez le champ Connexion vide.
   c. Pour explorer les données, choisissez le chemin spécifié dans mon compte.
   d. Pour Inclure le chemin, entrez le chemin S3 complet du dossier de sortie des sentiments :`s3://amzn-s3-demo-bucket/sentiment-results`.
   e. Choisissez Suivant.
7. Pour Ajouter un autre magasin de données, cliquez sur Oui, puis sur Suivant. Répétez l'étape 6, mais entrez le chemin S3 complet du dossier de sortie des entités :`s3://amzn-s3-demo-bucket/entities-results`.
8. Pour Ajouter un autre magasin de données, cliquez sur Non, puis sur Suivant.
9. Pour Choisir un rôle IAM, procédez comme suit :

   a. Choisissez Créer un rôle IAM.
   b. Pour le rôle IAM, entrez `glue-access-role` puis choisissez Next.
10. Pour Créer un calendrier pour ce robot d'exploration, choisissez Exécuter à la demande, puis Suivant.
11. Pour configurer la sortie du robot d'exploration, procédez comme suit :

    a. Pour Base de données, sélectionnez Ajouter une base de données.
    b. Pour Database name (Nom de la base de données), entrez `comprehend-results`. Cette base de données stockera vos tables de sortie Amazon Comprehend.
    c. Conservez les paramètres par défaut des autres options et choisissez Next.
12. Passez en revue les informations du crawler, puis choisissez Terminer.
13. Dans la console Glue, dans Crawlers, choisissez `comprehend-analysis-crawler` Run crawler. La fin du robot d'exploration peut prendre quelques minutes.

Chargez les données dans un AWS Glue Data Catalog (AWS CLI)

Créez un rôle IAM AWS Glue qui autorise l'accès à votre compartiment S3. Créez ensuite une base de données dans le AWS Glue Data Catalog. Enfin, créez et exécutez un robot d'exploration qui charge vos données dans les tables de la base de données.

Pour charger les données dans un AWS Glue Data Catalog (AWS CLI)

1. Pour créer un rôle IAM pour AWS Glue, procédez comme suit :

   a. Enregistrez la politique de confiance suivante sous forme de document JSON appelé `glue-trust-policy.json` sur votre ordinateur.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "glue.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

   b. Pour créer un rôle IAM, exécutez la commande suivante. *path/*Remplacez-le par le chemin de votre ordinateur local vers le document JSON.

   ```
   aws iam create-role --role-name glue-access-role
   --assume-role-policy-document file://path/glue-trust-policy.json
   ```

   c. Lorsque le numéro de ressource Amazon (ARN) du nouveau rôle est AWS CLI répertorié, copiez-le et enregistrez-le dans un éditeur de texte.

   d. Enregistrez la politique IAM suivante sous forme de document JSON appelé `glue-access-policy.json` sur votre ordinateur. La politique AWS Glue autorise l'exploration de vos dossiers de résultats.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
   ```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/sentiment-results*",
                "arn:aws:s3:::amzn-s3-demo-bucket/entities-results*"
            ]
        }
    ]
}
```

e. Pour créer la politique IAM, exécutez la commande suivante. *path/*Remplacez-le par le chemin de votre ordinateur local vers le document JSON.

```
aws iam create-policy --policy-name glue-access-policy
--policy-document file://path/glue-access-policy.json
```

f. Lorsque l'ARN de la politique d'accès est AWS CLI répertorié, copiez-le et enregistrez-le dans un éditeur de texte.

g. Associez la nouvelle politique au rôle IAM en exécutant la commande suivante. *policy-arn*Remplacez-le par l'ARN de la politique IAM que vous avez copié à l'étape précédente.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name glue-access-role
```

h. Associez la politique AWS gérée `AWSGlueServiceRole` à votre rôle IAM en exécutant la commande suivante.

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
--role-name glue-access-role
```

2. Créez une AWS Glue base de données en exécutant la commande suivante.

```
aws glue create-database
--database-input Name="comprehend-results"
```

3. Créez un nouveau AWS Glue crawler en exécutant la commande suivante. *glue-iam-role-arn*Remplacez-le par l'ARN de votre rôle AWS Glue IAM.

```
aws glue create-crawler
--name comprehend-analysis-crawler
--role glue-iam-role-arn
--targets S3Targets=[
{Path="s3://amzn-s3-demo-bucket/sentiment-results"},
{Path="s3://amzn-s3-demo-bucket/entities-results"}]
--database-name comprehend-results
```

4. Démarrez le crawler en exécutant la commande suivante.

```
aws glue start-crawler --name comprehend-analysis-crawler
```

La fin du robot d'exploration peut prendre quelques minutes.

Préparer les données pour l'analyse

Vous disposez à présent d'une base de données contenant les résultats d'Amazon Comprehend. Cependant, les résultats sont imbriqués. Pour les désimbriquer, vous devez exécuter quelques instructions SQL dans Amazon Athena. Amazon Athena est un service de requête interactif qui facilite l'analyse des données dans Amazon S3 à l'aide du SQL standard. Athena fonctionne sans serveur, il n'y a donc aucune infrastructure à gérer et dispose d'un pay-per-query modèle de tarification. Au cours de cette étape, vous créez de nouvelles tables de données nettoyées que vous pouvez utiliser pour l'analyse et la visualisation. Vous utilisez la console Athena pour préparer les données.

Pour préparer les données

1. Ouvrez la console à l'adresse https://console.aws.amazon.com/athena/.
2. Dans l'éditeur de requête, choisissez Settings (Paramètres), puis Manage (Gérer).
3. Pour Emplacement des résultats de la requête, entrez`s3://amzn-s3-demo-bucket/query-results/`. Cela crée un nouveau dossier appelé `query-results` dans votre compartiment qui stocke le résultat des Amazon Athena requêtes que vous exécutez. Choisissez Save (Enregistrer).
4. Dans l'éditeur de requêtes, sélectionnez Editeur.
5. Pour Base de données, choisissez la AWS Glue base de données `comprehend-results` que vous avez créée.

6. Dans la section Tables, vous devriez avoir deux tables appelées `sentiment_results` et`entities_results`. Prévisualisez les tableaux pour vous assurer que le robot a chargé les données. Dans les options de chaque tableau (les trois points à côté du nom du tableau), choisissez Aperçu du tableau. Une courte requête s'exécute automatiquement. Vérifiez le volet Résultats pour vous assurer que les tables contiennent des données.

   > **Tip**
   >
   > Si les tables ne contiennent aucune donnée, essayez de vérifier les dossiers de votre compartiment S3. Assurez-vous qu'il existe un dossier pour les résultats des entités et un dossier pour les résultats des sentiments. Essayez ensuite d'exécuter un nouveau AWS Glue robot d'exploration.

7. Pour désimbriquer la `sentiment_results` table, entrez la requête suivante dans l'éditeur de requêtes et choisissez Exécuter.

```
CREATE TABLE sentiment_results_final AS
SELECT file, line, sentiment,
sentimentscore.mixed AS mixed,
sentimentscore.negative AS negative,
sentimentscore.neutral AS neutral,
sentimentscore.positive AS positive
FROM sentiment_results
```

8. Pour commencer à dénicher la table des entités, entrez la requête suivante dans l'éditeur de requêtes et choisissez Exécuter.

```
CREATE TABLE entities_results_1 AS
SELECT file, line, nested FROM entities_results
CROSS JOIN UNNEST(entities) as t(nested)
```

9. Pour terminer la désimbrication de la table des entités, entrez la requête suivante dans l'éditeur de requêtes et choisissez Exécuter la requête.

```
CREATE TABLE entities_results_final AS
SELECT file, line,
nested.beginoffset AS beginoffset,
nested.endoffset AS endoffset,
nested.score AS score,
nested.text AS entity,
```

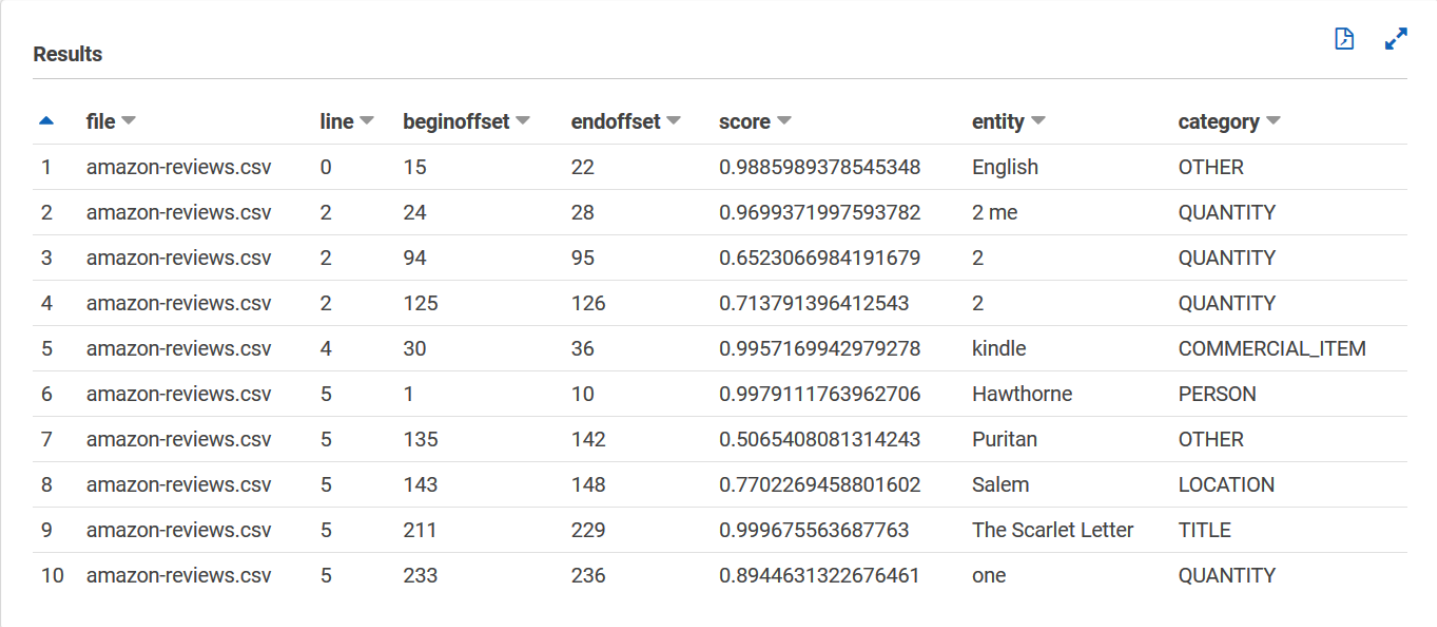
```
nested.type AS category
FROM entities_results_1
```

Votre `sentiment_results_final` tableau doit ressembler à ce qui suit, avec des colonnes nommées fichier, ligne, sentiment, mixte, négatif, neutre et positif. Le tableau doit comporter une valeur par cellule. La colonne des sentiments décrit le sentiment général le plus probable à l'égard d'un avis donné. Les colonnes mixtes, négatives, neutres et positives donnent des scores pour chaque type de sentiment.

Results

	file	line	sentiment	mixed	negative	neutral	positive
1	amazon-reviews.csv	6	MIXED	0.9862896203994751	0.0015502438182011247	1.6660270921420306E-4	0.0119935879483
2	amazon-reviews.csv	8	POSITIVE	0.0012987082591280341	0.01186690479516983	0.174478679895401	0.8123556375503
3	amazon-reviews.csv	11	POSITIVE	6.5368581090297084E-6	0.0013866390800103545	0.007405391428619623	0.9912014007568
4	amazon-reviews.csv	13	POSITIVE	4.7155481297522783E-4	0.24615342915058136	0.017713148146867752	0.7356618046760
5	amazon-reviews.csv	14	POSITIVE	1.5821871784282848E-5	0.06828905642032623	0.014075091108679771	0.9176200628280
6	amazon-reviews.csv	16	MIXED	0.9864791035652161	8.548551704734564E-4	1.0789262159960344E-4	0.0125581491738
7	amazon-reviews.csv	20	NEGATIVE	1.1621621524682269E-4	0.9815887212753296	0.004688907880336046	0.0136061981320
8	amazon-reviews.csv	21	POSITIVE	4.663573781726882E-5	0.009533549658954144	0.0015825830632820725	0.9888372421264
9	amazon-reviews.csv	23	POSITIVE	1.7699007003102452E-4	0.40269607305526733	0.0018250439316034317	0.5953019261360
10	amazon-reviews.csv	25	POSITIVE	1.8434448065818287E-6	1.15832663141191E-4	0.0010993879986926913	0.9987829327583

Votre **`entities_results_final`** tableau doit ressembler à ce qui suit, avec des colonnes nommées file, line, beginoffset, endoffset, score, entity et category. Le tableau doit comporter une valeur par cellule. La colonne de score indique la confiance d'Amazon Comprehend dans l'entité détectée. La catégorie indique le type d'entité détecté par Comprehend.



Results

	file	line	beginoffset	endoffset	score	entity	category
1	amazon-reviews.csv	0	15	22	0.9885989378545348	English	OTHER
2	amazon-reviews.csv	2	24	28	0.9699371997593782	2 me	QUANTITY
3	amazon-reviews.csv	2	94	95	0.6523066984191679	2	QUANTITY
4	amazon-reviews.csv	2	125	126	0.713791396412543	2	QUANTITY
5	amazon-reviews.csv	4	30	36	0.9957169942979278	kindle	COMMERCIAL_ITEM
6	amazon-reviews.csv	5	1	10	0.9979111763962706	Hawthorne	PERSON
7	amazon-reviews.csv	5	135	142	0.5065408081314243	Puritan	OTHER
8	amazon-reviews.csv	5	143	148	0.7702269458801602	Salem	LOCATION
9	amazon-reviews.csv	5	211	229	0.999675563687763	The Scarlet Letter	TITLE
10	amazon-reviews.csv	5	233	236	0.8944631322676461	one	QUANTITY

Maintenant que les résultats d'Amazon Comprehend sont chargés dans des tableaux, vous pouvez visualiser et extraire des informations pertinentes à partir des données.

Étape 5 : Visualisation du résultat d'Amazon Comprehend dans QuickSight

Après avoir enregistré les résultats d'Amazon Comprehend dans des tableaux, vous pouvez vous connecter aux données et les visualiser avec. QuickSight QuickSight est un outil de business intelligence (BI) AWS géré pour visualiser les données. QuickSightvous permet de vous connecter facilement à votre source de données et de créer de puissants visuels. Au cours de cette étape, vous vous connectez QuickSight à vos données, créez des visualisations qui extraient des informations à partir des données et publiez un tableau de bord des visualisations.

Rubriques

- Prérequis
- Donne à QuickSight un accès
- Importer les ensembles de données
- Créez une visualisation des sentiments
- Création d'une visualisation d'entités
- Publier un tableau de bord
- Nettoyage

Prérequis

Avant de commencer, complétez Étape 4 : Préparation de la sortie Amazon Comprehend pour la visualisation des données.

Donne à QuickSight un accès

Pour importer les données, vous devez avoir QuickSight accès à votre compartiment Amazon Simple Storage Service (Amazon Simple Storage Service (Amazon S3 Amazon Athena) et à vos tables. Pour donner QuickSight accès à vos données, vous devez être connecté en tant qu' QuickSight administrateur et être autorisé à modifier les autorisations relatives aux ressources. Si vous ne parvenez pas à effectuer les étapes suivantes, passez en revue les prérequis IAM sur la page de présentation. Tutoriel : Analyse des informations issues des avis clients avec Amazon Comprehend

Pour donner QuickSight accès à vos données

1. Ouvrez la QuickSight console.
2. Si c'est la première fois que vous l'utilisez QuickSight, la console vous invite à créer un nouvel utilisateur administrateur en fournissant une adresse e-mail. Pour Adresse e-mail, entrez la même adresse e-mail que votre Compte AWS. Choisissez Continuer.
3. Une fois connecté, choisissez le nom de votre profil dans la barre de navigation, puis sélectionnez Gérer QuickSight. Vous devez être connecté en tant qu'administrateur pour voir l' QuickSightoption Gérer.
4. Choisissez Sécurité et autorisations.
5. Pour QuickSight accéder aux AWS services, choisissez Ajouter ou supprimer.
6. Choisissez Amazon S3.
7. Dans Select Amazon S3 buckets, choisissez votre compartiment S3 pour les autorisations S3 Bucket et Write pour Athena Workgroup.
8. Choisissez Finish (Terminer).
9. Choisissez Mettre à jour.

Importer les ensembles de données

Avant de créer des visualisations, vous devez y ajouter les ensembles de données de sentiments et d'entités. QuickSight Vous pouvez le faire avec la QuickSight console. Vous importez vos tables de sentiments non imbriqués et d'entités non imbriquées à partir de. Amazon Athena

Pour importer vos ensembles de données

1. Ouvrez la QuickSight console.
2. Dans la barre de navigation, dans Ensembles de données, sélectionnez Nouvel ensemble de données.
3. Pour Créer un ensemble de données, choisissez Athena.
4. Dans le champ Nom de la source de données, entrez `reviews-sentiment-analysis` et choisissez Créer une source de données.
5. Pour Database (Base de données), choisissez la base de données `comprehend-results`.
6. Pour les tableaux, choisissez le tableau `sentiment_results_final` des sentiments, puis sélectionnez Sélectionner.
7. Choisissez Importer vers SPICE pour des analyses plus rapides, puis choisissez Visualize. SPICE est QuickSight le moteur de calcul en mémoire qui fournit des analyses plus rapides que les requêtes directes lors de la création de visualisations.
8. Retournez à la QuickSight console et choisissez Datasets. Répétez les étapes 1 à 7 pour créer un jeu de données d'entités, mais apportez les modifications suivantes :

    a. Dans Nom de la source de données, entrez`reviews-entities-analysis`.

    b. Pour Tables, choisissez la table des entités`entities_results_final`.

Créez une visualisation des sentiments

Maintenant que vous pouvez accéder à vos données QuickSight, vous pouvez commencer à créer des visualisations. Vous créez un graphique circulaire avec les données de sentiment d'Amazon Comprehend. Le diagramme circulaire indique la proportion d'avis positifs, neutres, mitigés et négatifs.

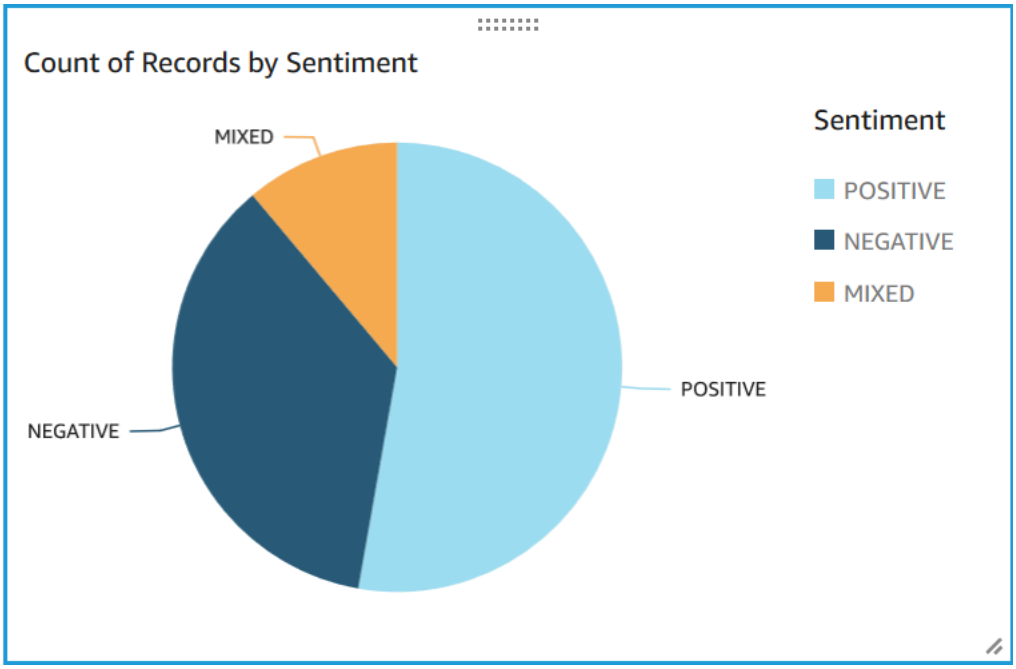
Pour visualiser les données relatives aux sentiments

1. Dans la QuickSight console, choisissez Analyses, puis choisissez Nouvelle analyse.
2. Dans Vos ensembles de données, choisissez l'ensemble de données de sentiments, `sentiment_results_final` puis sélectionnez Créer une analyse.
3. Dans l'éditeur visuel, dans la liste des champs, choisissez sentiment.

> **Note**
>
> Les valeurs de la liste des champs dépendent des noms de colonnes que vous avez utilisés pour créer les tables Amazon Athena. Si vous avez modifié les noms de colonnes fournis dans les requêtes SQL, les noms des listes de champs seront différents de ceux utilisés dans ces exemples de visualisation.

4. Pour les types visuels, choisissez Graphique à secteurs.

Un graphique circulaire similaire au suivant avec des sections positives, neutres, mixtes et négatives s'affiche. Pour voir le nombre et le pourcentage d'une section, passez la souris dessus.



Création d'une visualisation d'entités

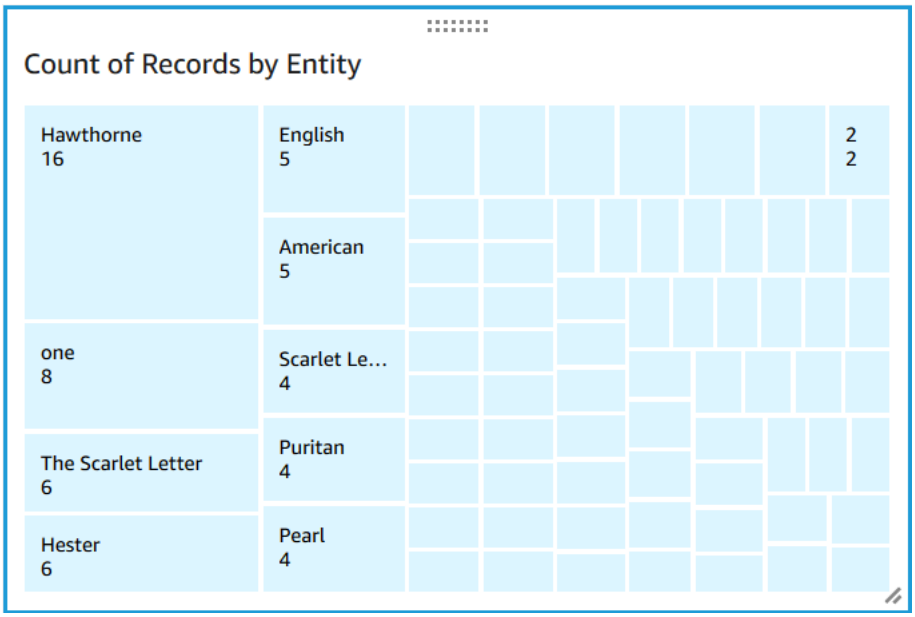
Créez maintenant une deuxième visualisation avec le jeu de données des entités. Vous créez une arborescence des différentes entités présentes dans les données. Chaque bloc de l'arborescence représente une entité, et la taille du bloc est corrélée au nombre de fois que l'entité apparaît dans le jeu de données.

Pour visualiser les données des entités

1. Dans le volet de configuration Visualize, à côté de Ensemble de données, cliquez sur l'icône Ajouter, modifier, remplacer et supprimer des ensembles de données.

2. Choisissez Ajouter un jeu de données.
3. Pour Choisir un ensemble de données à ajouter, choisissez votre jeu `entities_results_final` de données d'entités dans la liste des jeux de données, puis sélectionnez Sélectionner.
4. Dans le volet de configuration Visualize, choisissez le menu déroulant Ensemble de données et choisissez le jeu de données des entités`entities_results_final`.
5. Dans la liste des champs, sélectionnez une entité.
6. Pour les types visuels, choisissez Carte arborescente.

Une arborescence similaire à la suivante s'affiche à côté de votre graphique circulaire. Pour voir le nombre d'entités spécifiques, passez le curseur sur un bloc.



Publier un tableau de bord

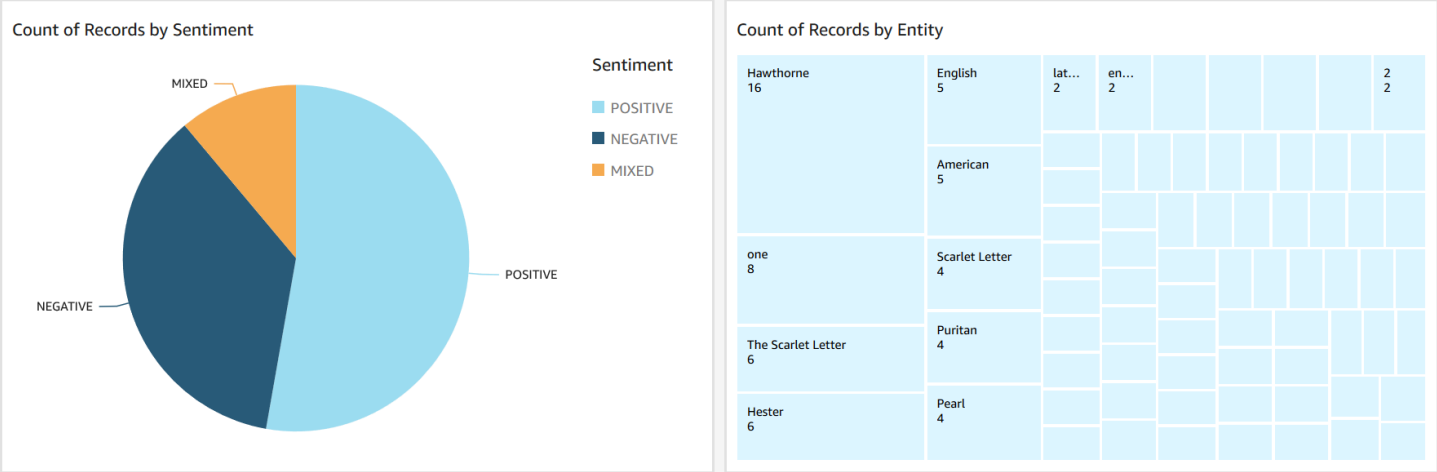
Après avoir créé les visualisations, vous pouvez les publier sous forme de tableau de bord. Vous pouvez effectuer diverses tâches à l'aide d'un tableau de bord, comme le partager avec les utilisateurs Compte AWS, l'enregistrer au format PDF ou l'envoyer par e-mail sous forme de rapport (limité à l'édition Enterprise de QuickSight). Au cours de cette étape, vous publiez les visualisations sous forme de tableau de bord dans votre compte.

Pour publier votre tableau de bord

1. Dans la barre de navigation, choisissez Partager.

2. Choisissez Publish dashboard (Publier le tableau de bord).
3. Choisissez Publier le nouveau tableau de bord sous et entrez le nom `comprehend-analysis-reviews` du tableau de bord.
4. Choisissez Publish dashboard (Publier le tableau de bord).
5. Fermez le panneau Partager le tableau de bord avec les utilisateurs en cliquant sur le bouton de fermeture situé dans l'angle supérieur droit.
6. Dans la QuickSight console, dans le panneau de navigation, choisissez Tableaux de bord. Une miniature de votre nouveau tableau de bord **`comprehend-analysis-reviews`** devrait apparaître sous Tableaux de bord. Sélectionnez le tableau de bord pour l'afficher.

Vous avez maintenant un tableau de bord avec des visualisations de sentiments et d'entités qui ressemble à l'exemple suivant.



> **Tip**
>
> Si vous souhaitez modifier les visualisations de votre tableau de bord, retournez dans Analyses et modifiez la visualisation que vous souhaitez mettre à jour. Publiez ensuite à nouveau le tableau de bord, soit en tant que nouveau tableau de bord, soit en remplacement du tableau de bord existant.

Nettoyage

Après avoir terminé ce tutoriel, vous voudrez peut-être nettoyer toutes les AWS ressources que vous ne voulez plus utiliser. Les AWS ressources actives peuvent continuer à être facturées sur votre compte.

Les mesures suivantes peuvent aider à éviter d'encourir des frais permanents :

- Annulez votre QuickSight abonnement. QuickSight est un service d'abonnement mensuel. Pour annuler votre abonnement, consultez la section Annulation de votre abonnement dans le guide de l'QuickSight utilisateur.
- Supprimer votre compartiment Amazon S3. Amazon S3 vous facture le stockage. Pour nettoyer vos ressources Amazon S3, supprimez votre compartiment. Pour plus d'informations sur la suppression d'un compartiment, voir Comment supprimer un compartiment S3 ? dans le guide de l'utilisateur d'Amazon Simple Storage Service. Assurez-vous d'enregistrer tous vos fichiers importants avant de supprimer votre bucket.
- Effacez votre AWS Glue Data Catalog. Le AWS Glue Data Catalog stockage vous est facturé chaque mois. Vous pouvez supprimer vos bases de données pour éviter d'encourir des frais permanents. Pour plus d'informations sur la gestion de vos AWS Glue Data Catalog bases de données, consultez la section Utilisation des bases de données sur la AWS Glue console dans le manuel du AWS Glue développeur. Assurez-vous d'exporter vos données avant de supprimer des bases de données ou des tables.

Utilisation des points d'accès Lambda des objets Amazon S3 pour les informations personnelles identifiables (PII)

Utilisez les points d'accès Amazon S3 Object Lambda pour les informations personnelles identifiables (PII) afin de configurer la manière dont les documents sont extraits de votre compartiment Amazon S3. Vous pouvez contrôler l'accès aux documents contenant des informations personnelles et supprimer des informations personnelles des documents. Pour plus d'informations sur la manière dont Amazon Comprehend peut détecter les informations personnelles dans vos documents, consultez. Détection des entités PII Les points d'accès Amazon S3 Object Lambda utilisent AWS Lambda des fonctions pour transformer automatiquement le résultat d'une requête GET Amazon S3 standard. Pour plus d'informations, consultez la section Transformation d'objets avec l'objet S3 Lambda dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Lorsque vous créez un point d'accès Amazon S3 Object Lambda pour les informations personnelles, les documents sont traités à l'aide des fonctions Amazon Comprehend Lambda pour contrôler l'accès aux documents contenant des informations personnelles et supprimer les informations personnelles des documents.

Lorsque vous créez un point d'accès Amazon S3 Object Lambda pour les informations personnelles, les documents sont traités à l'aide des fonctions Amazon Comprehend Lambda suivantes :

- `ComprehendPiiAccessControlS3ObjectLambda`- Contrôle l'accès aux documents contenant des informations personnelles stockées dans votre compartiment S3. Pour plus d'informations sur cette fonction Lambda, connectez-vous au AWS Management Console pour voir la ObjectLambda fonction ComprehendPiiAccessControlS3 dans le. AWS Serverless Application Repository
- `ComprehendPiiRedactionS3ObjectLambda`- Supprime les informations personnelles des documents de votre compartiment Amazon S3. Pour plus d'informations sur cette fonction Lambda, connectez-vous au AWS Management Console pour voir la ObjectLambda fonction ComprehendPiiRedactionS3 dans le. AWS Serverless Application Repository

Pour plus d'informations sur le déploiement d'applications sans serveur depuis le AWS Serverless Application Repository, consultez la section Déploiement d'applications dans le AWS Serverless Application Repository Developer Guide.

Rubriques

- Contrôle de l'accès aux documents contenant des informations personnelles identifiables (PII)
- Supprimer des informations personnelles identifiables (PII) à partir de documents

Contrôle de l'accès aux documents contenant des informations personnelles identifiables (PII)

Vous pouvez utiliser un point d'accès Amazon S3 Object Lambda pour contrôler l'accès aux documents contenant des informations personnelles identifiables (PII).

Pour garantir que seuls les utilisateurs autorisés ont accès aux documents contenant des informations personnelles stockées dans votre compartiment Amazon S3, vous utilisez cette `ComprehendPiiAccessControlS3ObjectLambda` fonction. Cette fonction Lambda utilise l'ContainsPiiEntitiesopération lors du traitement d'une requête GET Amazon S3 standard sur des objets de document.

Par exemple, si votre compartiment S3 contient des documents contenant des informations personnelles telles que des numéros de carte de crédit ou des informations de compte bancaire, vous pouvez configurer la `ComprehendPiiAccessControlS3ObjectLambda` fonction pour détecter

ces types d'entités PII et restreindre l'accès aux utilisateurs non autorisés. Pour plus d'informations sur les types d'entités PII pris en charge, consultezTypes d'entités universelles PII.

Pour plus d'informations sur cette fonction Lambda, connectez-vous au AWS Management Console pour voir la ObjectLambda fonction ComprehendPiiAccessControlS3 dans le. AWS Serverless Application Repository

Création d'un point d'accès Lambda aux objets Amazon S3 pour contrôler l'accès aux documents

L'exemple suivant crée un point d'accès Amazon S3 Object Lambda pour contrôler l'accès aux documents contenant des numéros de sécurité sociale.

Création d'un point d'accès Lambda à un objet Amazon S3 à l'aide du AWS Command Line Interface

Créez une configuration de point d'accès Amazon S3 Object Lambda et enregistrez-la dans un fichier appelé config.json.

```
{
    "SupportingAccessPoint": "s3-default-access-point-name-arn",
    "TransformationConfigurations": [
        {
            "Actions": [
                "s3:GetObject"
            ],
            "ContentTransformation": {
                "AwsLambda": {
                    "FunctionArn": "comprehend-pii-access-control-s3-object-lambda-
arn",
                    "FunctionPayload": "{\"pii_entities_types\": \"SSN\"}"
                }
            }
        }
    ]
}
```

L'exemple suivant crée un point d'accès Amazon S3 Object Lambda en fonction de la configuration définie dans le `config.json` fichier.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws s3control create-banner-access-point \
    --region region \
    --account-id account-id \
    --name s3-object-lambda-access-point \
    --configuration file://config.json
```

Invocation d'un point d'accès Lambda à un objet Amazon S3 pour contrôler l'accès aux documents

L'exemple suivant invoque un point d'accès Lambda aux objets Amazon S3 pour contrôler l'accès aux documents.

Invocation d'un point d'accès Lambda à un objet Amazon S3 à l'aide du AWS Command Line Interface

L'exemple suivant invoque un point d'accès Lambda d'objets Amazon S3 à l'aide du. AWS CLI

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws s3api get-object \
    --region region \
    --bucket s3-object-lambda-access-point-name-arn \
    --key object-prefix-key output-file-name
```

Supprimer des informations personnelles identifiables (PII) à partir de documents

Vous pouvez utiliser un point d'accès Amazon S3 Object Lambda pour supprimer les informations personnelles identifiables (PII) des documents.

Pour supprimer les types d'entités PII des documents stockés dans un compartiment S3, vous utilisez la `ComprehendPiiRedactionS3ObjectLambda` fonction. Cette fonction Lambda utilise les DetectPiiEntitiesopérations ContainsPiiEntitiesand lors du traitement d'une requête GET Amazon S3 standard sur des objets de document.

Par exemple, si les documents de votre compartiment S3 contiennent des informations personnelles telles que des numéros de carte de crédit ou des informations de compte bancaire, vous pouvez

configurer la `ComprehendPiiRedactionS3ObjectLambda` fonction pour détecter les informations personnelles, puis renvoyer une copie de ces documents dans lesquels les types d'entités PII sont expurgés. Pour plus d'informations sur les types d'entités PII pris en charge, consultezTypes d'entités universelles PII.

Pour plus d'informations sur cette fonction Lambda, connectez-vous au AWS Management Console pour voir la ObjectLambda fonction ComprehendPiiRedactionS3 dans le. AWS Serverless Application Repository

Création d'un point d'accès Lambda à un objet Amazon S3 pour supprimer les informations personnelles des documents

L'exemple suivant crée un point d'accès Amazon S3 Object Lambda pour récupérer les numéros de carte de crédit des documents.

Création d'un point d'accès Lambda à un objet Amazon S3 à l'aide du AWS Command Line Interface

Créez une configuration de point d'accès Amazon S3 Object Lambda et enregistrez-la dans un fichier appelé. `config.json`

```
{
    "SupportingAccessPoint": "s3-default-access-point-name-arn",
    "TransformationConfigurations": [
        {
            "Actions": [
                "s3:GetObject"
            ],
            "ContentTransformation": {
                "AwsLambda": {
                    "FunctionArn": "comprehend-pii-redaction-s3-object-lambda-arn",
                    "FunctionPayload": "{\"pii_entities_types\": \"CREDIT_DEBIT_NUMBER
\"}"
                }
            }
        }
    ]
}
```

L'exemple suivant illustre la création d'un point d'accès Lambda aux objets Amazon S3 sur la base de la configuration définie dans le `config.json`

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws s3control create-access-point-for-object-lambda \
    --region region \
    --account-id account-id \
    --name s3-object-lambda-access-point \
    --configuration file://config.json
```

Invocation d'un point d'accès Lambda à un objet Amazon S3 pour supprimer des informations personnelles contenues dans des documents

Les exemples suivants invoquent un point d'accès Amazon S3 Object Lambda pour supprimer les informations personnelles des documents.

Invocation d'un point d'accès Lambda à un objet Amazon S3 à l'aide du AWS Command Line Interface

L'exemple suivant invoque un point d'accès Lambda d'objets Amazon S3 à l'aide du. AWS CLI

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix, à savoir la barre oblique inversée (\), à la fin de chaque ligne par un accent circonflexe (^).

```
aws s3api get-object \
    --region region \
    --bucket s3-object-lambda-access-point-name-arn \
    --key object-prefix-key output-file-name
```

Solution : analyser du texte avec Amazon Comprehend et OpenSearch

AWS fournit une implémentation de référence de l'analyse de texte à l'aide d'Amazon Comprehend et du OpenSearch service. Amazon Comprehend fournit une analyse de texte ainsi que l'indexation, la recherche et OpenSearch la visualisation de documents.

Pour plus d'informations, consultez Analyser du texte avec Amazon Comprehend OpenSearch et Amazon Comprehend.

Référence d’API

La référence de l'API est désormais un document distinct. Pour plus d'informations, consultez le manuel Amazon Comprehend API Reference.

Historique du document pour Amazon Comprehend

Le tableau suivant décrit la documentation de cette version d'Amazon Comprehend.

Modification	Description	Date
Formation personnalisée sur les classificateurs avec des documents natifs	Amazon Comprehend prend désormais en charge la formation personnalisée aux classificateurs avec des documents natifs. Pour plus d'informations, consultez la section Modèles de classification des formations dans Amazon Comprehend.	19 avril 2023
Volants pour gérer des modèles personnalisés	Amazon Comprehend prend désormais en charge les volants pour vous aider à gérer la formation et le suivi des versions des modèles pour les modèles personnalisés. Pour plus d'informations, consultez Flywheels dans Amazon Comprehend.	28 février 2023
Rubriques de sécurité IAM mises à jour	Les rubriques relatives à la sécurité IAM ont été mises à jour pour inclure les identités fédérées. Pour plus d'informations, consultez Identity and Access Management pour Amazon Comprehend.	22 décembre 2022
Traitement en une étape pour l'inférence avec des modèles personnalisés	Amazon Comprehend extrait désormais automatiquement le texte des documents	1er décembre 2022

	d'entrée image, PDF ou Word avant d'exécuter une classification personnalisée ou une reconnaissance d'entité personnalisée. Pour plus d'informations, consultez la section Traitement des documents dans Amazon Comprehend.	
Synchrone APIs pour un sentiment ciblé	Amazon Comprehend prend désormais en charge l'analyse synchrone APIs et en temps réel sur console pour cibler les sentiments. Le sentiment ciblé détermine le sentiment associé à des entités spécifiques dans un document. Pour plus d'informations, consultez la section Sentiment ciblé dans Amazon Comprehend.	21 septembre 2022
Réduction du nombre minimum d'annotations pour les personnes chargées de la reconnaissance des formations	Amazon Comprehend a réduit les exigences minimales pour former un outil de reconnaissance à l'aide de fichiers d'annotation CSV en texte brut. Vous pouvez désormais créer un modèle de reconnaissance d'entité personnalisé avec seulement trois documents annotés et au moins 25 annotations par type d'entité. Pour plus d'informations, consultez la section Préparation des données d'entraînement.	3 août 2022

Taille du document d'entrée accrue pour le temps réel APIs	Amazon Comprehend prend désormais en charge un document d'entrée allant jusqu'à 100 Ko pour la plupart en temps réel. APIs Pour plus d'informations, consultez la section Directives et quotas.	18 juillet 2022
Types d'entités PII supplémentaires	Des types d'entités PII supplémentaires sont désormais détectés par Amazon Comprehend. Pour plus d'informations, consultez la section Détection des entités PII dans Amazon Comprehend.	20 mai 2022
Restructuration de la table des matières	Restructuration de la table des matières du guide du développeur Amazon Comprehend pour faciliter la navigation. Pour plus d'informations, consultez What is Amazon Comprehend.	7 avril 2022
Sentiment ciblé	Amazon Comprehend prend désormais en charge l'analyse ciblée des sentiments, qui détermine le sentiment associé à des entités spécifiques dans un document. Pour plus d'informations, consultez la section Sentiment ciblé dans Amazon Comprehend.	9 mars 2022

Nouvelle fonction	Amazon Comprehend vous permet désormais d'analyser des images pour une reconnaissance personnalisée des entités. Pour plus d'informations, consultez la section Détection d'entités personnalisées dans Amazon Comprehend.	28 février 2022
Nouvelle fonction	Vous pouvez désormais copier des modèles personnalisés entraînés entre deux Comptes AWS. Pour plus d'informations, consultez Copier des modèles personnalisés entre comptes dans Amazon Comprehend.	2 février 2022
Nouvelle fonction	Vous pouvez désormais consulter les AWS Trusted Advisor recommandations qui peuvent vous aider à optimiser le coût et la sécurité de vos points de terminaison Amazon Comprehend. Pour plus d'informations, consultez Utilisation Trusted Advisor avec Amazon Comprehend.	29 septembre 2021

Nouvelle fonction	Amazon Comprehend a lancé une suite de fonctionnalités pour Comprehend Custom qui permet d'améliorer continuel lement les modèles en vous donnant la possibilité de créer de nouvelles versions de modèles, de les tester en continu sur des ensembles de tests spécifiques et d'effectuer une migration en direct vers de nouveaux points de terminais on du modèle.	21 septembre 2021
Nouvelle fonction	Amazon Comprehend vous permet désormais d'analyser des documents PDF et Word pour une reconnaissance personnalisée des entités. Avec les formats PDF et Word, vous pouvez extraire des informations de documents contenant des en-têtes, des listes et des tableaux.	14 septembre 2021

Nouvelle fonction	Amazon Comprehend a lancé une nouvelle fonctionnalité de présentation des points de terminaison qui vous fournit une vue globale de vos points de terminaison. Sur la page de présentation des terminaux , vous pouvez consulter tous vos terminaux en un seul endroit pour comprendre l'utilisation de vos terminaux par rapport à votre utilisation réelle des ressources.	24 août 2021
Nouvelle fonction	Amazon Comprehend Medical vous permet désormais d'établir une connexion privée avec votre Virtual Private Cloud (VPC) en créant un point de terminaison VPC d'interface. Pour plus d'informations, consultez la section Points de terminaison VPC (). PrivateLink	13 juin 2021
Extension de la langue	Amazon Comprehend a ajouté quatre langues supplémentaires pour la fonctionnalité linguistique dominante : le haoussa (ha), le laotien (lo), le maltais (mt) et l'oromo (om). Pour plus d'informations, consultez la section Langues prises en charge dans Amazon Comprehend.	10 mai 2021

Nouvelle fonction	Avec Amazon Comprehend, vous pouvez désormais chiffrer des modèles personnalisés à l'aide d'une clé gérée par le client (CMK). Pour plus d'informations, consultez la section Chiffrement KMS dans Amazon Comprehend.	31 mars 2021
Nouvelle fonction	Vous pouvez désormais utiliser les points d'accès Amazon S3 Object Lambda pour configurer la manière dont les documents contenant des informations personnelles identifiables (PII) sont extraits de votre compartiment Amazon S3. Vous pouvez contrôler l'accès aux documents contenant des informations personnelles et supprimer des informations personnelles des documents. Pour plus d'informations, consultez Utilisation des points d'accès Lambda des objets Amazon S3 pour les informations personnelles identifiables (PII).	18 mars 2021

Nouvelle fonction	Vous pouvez désormais étiqueter un document avec des informations personnel les identifiables (PII). Amazon Comprehend peut analyser votre document pour détecter la présence d'informations personnelles et renvoyer les étiquettes des types d'entités à informations personnel les identifiés, tels que le nom, l'adresse, le numéro de compte bancaire ou le numéro de téléphone. Pour plus d'informations, voir Étiqueter le document avec des informations personnelles.	11 mars 2021
Nouvelle fonction	Avec Amazon Comprehend, vous pouvez désormais détecter des événements dans un ensemble de documents. Lorsque vous créez une tâche de détection d'événements asynchrones, Amazon Comprehend peut détecter les types d'événements financiers pris en charge. Pour plus d'informations, consultez la section Détecter des événements.	24 novembre 2020

Nouvelle fonction	Amazon Comprehend vous permet désormais d'utiliser le dimensionnement automatiq ue pour les points de terminais on personnalisés de reconnais sance d'entités. Grâce à la mise à l'échelle automatiq ue, vous pouvez configurer automatiquement le provision nement des terminaux en fonction de vos besoins en matière de capacité. Pour plus d'informations, consultez la section Mise à l'échelle automatique avec les points de terminaison.	28 septembre 2020
Nouvelle fonction	Pour former des classificateurs personnalisés ou des outils de reconnaissance d'entités, vous pouvez désormais fournir des fichiers manifestes augmentés , qui sont des ensembles de données étiquetés produits par Amazon AI Ground SageMaker Truth. Pour plus d'informations sur ces fichiers et pour des exemples, consultez les sections Mode multi-classes, Modemulti-éti quettes et Annotations.	22 septembre 2020

Nouveau tutoriel	Amazon Comprehend propose désormais un didacticiel qui vous guide à travers un flux de travail multiservices consistant à analyser les avis clients et à visualiser les résultats des analyses. Pour plus d'informations, voir Tutoriel : Analyse des informations issues des avis.	17 septembre 2020
Nouvelle fonction	Avec Amazon Comprehend, vous pouvez désormais détecter dans votre texte les entités contenant des informations personnelles identifiables (PII), telles que des adresses, des numéros de compte bancaire ou des numéros de téléphone. Amazon Comprehend peut indiquer l'emplacement de chaque entité PII dans votre texte, ou fournir une copie de votre texte dans laquelle les IPI sont expurgées. Pour plus d'informations, consultez la section Détecter les informations personnelles identifiables (PII).	17 septembre 2020

Nouvelle fonction	Auparavant, vous ne pouviez entraîner un modèle que sur un maximum de 12 entités personnalisées. Amazon Comprehend vous permet désormais d'entraîner un modèle sur un maximum de 25 entités personnalisées à la fois. Pour plus d'informations, consultez la section Reconnaissance d'entités personnalisée.	12 août 2020
Extension de la langue	Amazon Comprehend a ajouté cinq langues supplémentaires pour la fonctionnalité de reconnaissance d'entité personnalisée : allemand (de), espagnol (es), français (fr), italien (it) et portugais (pt). Pour plus d'informations, consultez la section Langues prises en charge dans Amazon Comprehend.	12 août 2020
Nouvelle fonction	Amazon Comprehend vous permet désormais d'établir une connexion privée avec votre Virtual Private Cloud (VPC) en créant un point de terminaison VPC d'interface. Pour plus d'informations, consultez la section Points de terminaison VPC ().AWS PrivateLink	11 août 2020

Nouvelle fonction	Avec Amazon Comprehend, vous pouvez désormais détecter rapidement des entités personnalisées dans des documents texte individuels en effectuant une analyse en temps réel. Pour plus d'informations, consultez la section Détection d'entités personnalisées en temps réel avec Amazon Comprehend.	9 juillet 2020
Nouvelle fonctionnalité ajoutée	Amazon Comprehend prend désormais en charge un second mode de classification personnalisée asynchrone pour les documents, qui offre une plus grande flexibilité lors de l'application de classes personnalisées aux documents. Alors que le mode multi-classes n'associe qu'une seule classe à chaque document, le nouveau mode multi-étiquettes peut en associer plusieurs. Par exemple, un film peut être classé à la fois comme science-fiction et action. Pour plus d'informations, voir Modes multi-classes et multi-étiquettes dans la classification personnalisée.	19 décembre 2019

Nouvelle fonctionnalité ajoutée	Amazon Comprehend prend désormais en charge la classification personnalisée en temps réel pour les documents contenant du texte non structuré. Les clients peuvent utiliser une classification personnalisée en temps réel pour comprendre, étiqueter et acheminer les informations en fonction de leurs propres règles commerciales, de manière synchrone. Pour plus d'informations, voir Analyse en temps réel avec classification personnalisée.	25 novembre 2019
Nouvelles langues ajoutées	Amazon Comprehend a ajouté six langues supplémentaires pour plusieurs de ses fonctionnalités : l'arabe (ar), l'hindi (hi), le japonais (ja), le coréen (ko), le chinois simplifié (zh) et le chinois traditionnel (zh-TW). Ces nouvelles langues ne sont prises en charge que pour les opérations Determine Sentiment, Detect Key Phrases et Detect Entities non personnalisées. Pour plus d'informations, consultez la section Langues prises en charge.	6 novembre 2019

Nouvelle fonction	Auparavant, vous ne pouviez entraîner un modèle que sur une seule entité personnalisée. Par conséquent, vous ne pouviez rechercher que cette seule entité à l'aide d'une opération de reconnaissance d'entité. Amazon Comprehend a changé cela et vous pouvez désormais entraîner un modèle sur un maximum de 12 entités personnalisées à la fois. Pour plus d'informations, voir Reconnaissance d'entités personnalisée	9 juillet 2019
Nouvelle fonction	Amazon Comprehend fournit désormais une matrice de confusion à plusieurs classes pour une capacité accrue d'analyse des métriques lors de la formation d'un classificateur personnalisé. Ceci est actuellement pris en charge à l'aide du APIs seul. Pour plus d'informations, consultez la section Ressources de balisage dans Amazon Comprehend	5 avril 2019

Nouvelle fonction	Amazon Comprehend fournit des balises pour les classific ateurs personnalisés et les outils de reconnaissance d'entités personnalisés, qui peuvent être utilisés comme métadonnées vous permettan t d'organiser, de filtrer et de contrôler l'accès à vos ressources avec un niveau de contrôle plus fin que jamais. Pour plus d'informa tions, consultez la section Ressources de balisage dans Amazon Comprehend	3 avril 2019
Nouvelle fonction	Amazon S3 vous permet déjà de chiffrer vos documents d'entrée, et Amazon Comprehend va encore plus loin. En utilisant votre propre clé KMS, vous pouvez non seulement chiffrer les résultats de sortie de votre tâche, mais également les données du volume de stockage attaché à l'instance de calcul qui traite la tâche d'analyse. Le résultat est end-to-end la sécurité. Pour plus d'informations, consultez la section Chiffrement KMS dans Amazon Comprehend	28 mars 2019

Nouvelle fonction	La reconnaissance personnalisée des entités étend les fonctionnalités d'Amazon Comprehend en vous permettant d'identifier les nouveaux types d'entités non pris en charge parmi les types d'entités génériques prédéfinis. Cela signifie que vous pouvez analyser des documents et extraire des entités telles que des codes de produit ou des entités spécifiques à l'entreprise qui répondent à vos besoins particuliers. Pour plus d'informations, voir Reconnaissance d'entités personnalisée	16 novembre 2018
Nouvelle fonction	Vous pouvez utiliser Amazon Comprehend pour créer vos propres modèles de classification personnalisée, en assignant un document à une classe ou à une catégorie. Pour plus d'informations, consultez la section Classification des documents.	15 novembre 2018
Expansion de région	Amazon Comprehend est désormais disponible en Europe (Francfort) (eu-central-1).	10 octobre 2018

Extension de la langue	Outre l'anglais et l'espagnol, Amazon Comprehend peut désormais examiner des documents en français, allemand, italien et portugais. Pour plus d'informations, consultez la section Langues prises en charge dans Amazon Comprehend.	10 octobre 2018
Expansion de région	Amazon Comprehend est désormais disponible en Asie-Pacifique (Sydney) (ap-southeast-2).	15 août 2018
Nouvelle fonction	Amazon Comprehend analyse désormais les documents pour découvrir la syntaxe d'un document et la partie du discours correspondant à chaque mot. Pour plus d'informations, consultez Syntaxe.	17 juillet 2018
Nouvelle fonction	Amazon Comprehend prend désormais en charge le traitement par lots asynchrone pour la détection du langage, des phrases clés, des entités et des sentiments. Pour plus d'informations, consultez la section Traitement par lots asynchrone.	27 juin 2018

Nouveau guide	Il s'agit de la première version du manuel Amazon Comprehend Developer Guide.	29 novembre 2017

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.