aws

Guia do Desenvolvedor

Amazon Comprehend

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

Amazon Comprehend: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Amazon Comprehend? 1
Insights do Amazon Comprehend 2
Amazon Comprehend Custom 2
Flywheels 3
Clusters de documentos (modelos de tópicos) 3
Exemplos 3
Benefícios 4
Preços do Amazon Comprehend 5
Você é um usuário iniciante do Amazon Comprehend? 5
Como funciona 6
Insights 6
Entidades 7
Eventos 9
Frases chave 17
Idioma dominante 18
Sentimento 25
Sentimento direcionado 26
Análise de sintaxe 43
Amazon Comprehend Custom 47
Modelagem de tópicos 48
Modos de processamento de documentos 52
Processamento de documento único 53
Processamento síncrono de vários documentos 53
Processamento em lote assíncrono 56
Idiomas compatíveis 57
Idiomas compatíveis 57
Idiomas compatíveis com atributos do Amazon Comprehend 58
Configuração 60
Inscreva-se para um Conta da AWS 60
Criar um usuário com acesso administrativo 60
Configure o AWS CLI 62
Conceder acesso programático 62
Introdução 65
Utilizar o console 67

Análise em tempo real 67
Entidades 68
Frases chave 69
Idioma 70
Informações de identificação pessoal (PII) 71
Sentimento 73
Sentimento direcionado 74
Sintaxe 76
Tarefas de análise (console) 77
Uso da API 81
Trabalhando com AWS SDKs 81
Análise em tempo real (API) 82
Detectando o idioma dominante 83
Detectando entidades nomeadas 84
Detectando frases-chave 85
Determinando o sentimento 86
Análise em tempo real de um sentimento direcionado 88
Detectando a sintaxe 90
Lote em tempo real APIs 92
Trabalhos de análise (API) 97
Insights do Amazon Comprehend 98
Sentimento direcionado 104
Detecção de eventos 105
Modelagem de tópicos 110
Confiança e segurança 114
Detecção de toxicidade 115
Detecção de conteúdo tóxico usando a API 116
Classificação de segurança imediata 118
Classificação de segurança imediata usando a API 119
Detecção e redação de PIIs 121
Informações de identificação pessoal (PII) 122
Detectar entidades de PII 122
Localizar entidades de PII 123
Redigir entidades de PII 124
Tipos de entidades universais de PII 124
Tipos de entidade de PII específicos de cada país 127

Rotular entidades de PII 129
Análise em tempo real (console) 130
Offsets 71
Rótulos 73
Tarefas de análise assíncrona (console) 133
Análise em tempo real (API) 135
Localizando entidades de PII em tempo real (API) 136
Rotular entidades de PII em tempo real (API) 137
Trabalhos de análise (API) 138
Localizar entidades de PII 138
Redigir entidades de PII 143
Processamento de documentos 148
Entradas para análise em tempo real 148
Documentos de texto simples 149
Documentos semiestruturados 149
Arquivos de imagem e arquivos PDF digitalizados 149
Saída do Amazon Textract 149
Tamanhos máximos de documentos para análise em tempo real 149
Erros em documentos semiestruturados 150
Entradas para análise assíncrona 151
Documentos de texto simples 151
Documentos semiestruturados 152
Arquivos de imagem e arquivos PDF digitalizados 153
Arquivos JSON de saída do Amazon Textract 153
Configurar opções de extração de texto 154
Práticas recomendadas para imagens 155
Classificação personalizada 157
Preparar os dados de treinamento 158
Formatos de arquivo de treinamento 159
Modo multiclasse 160
Modo multirrótulo 163
Modelos de classificação de treinamento 167
Treine classificadores personalizados (console) 168
Treine classificadores personalizados (API) 173
Teste os dados de treinamento 175
Saída do treinamento do classificador 176

Métricas ... 181
Executar análises em tempo real ... 186
Análise em tempo real (console) ... 186
Análise em tempo real (API) ... 189
Saídas para análise em tempo real ... 192
Executar trabalhos de análise assíncrona ... 194
Formatos de arquivos de entrada ... 194
Tarefas de análise (console) ... 196
Trabalhos de análise (API) ... 198
Saídas para trabalhos de análise ... 200
Reconhecimento de entidade personalizado ... 205
Preparar os dados de treinamento ... 206
Quando usar anotações em vez de listas de entidades ... 207
Listas de entidades ... 208
Anotações ... 211
Treinamento de modelos reconhecedores ... 225
Treinar reconhecedores personalizados (console) ... 226
Treinar reconhecedores personalizados (API) ... 232
Métricas ... 235
Executar análises em tempo real ... 238
Análise em tempo real (console) ... 239
Análise em tempo real (API) ... 241
Saídas para análise em tempo real ... 244
Executar trabalhos de análise assíncrona ... 251
Tarefas de análise (console) ... 252
Trabalhos de análise (API) ... 254
Saídas para trabalhos de análise ... 257
Gerenciar modelos personalizados ... 263
Versionamento do modelo com o Amazon Comprehend ... 263
Copiar modelos personalizados entre Contas da AWS ... 266
Como compartilhar um modelo personalizado ... 267
Como importar um modelo personalizado ... 277
Flywheels ... 285
Visão geral do flywheel ... 285
Conjuntos de dados do flywheel ... 286
Criação do flywheel ... 287

Estados do flywheel 287
Iterações do flywheel 288
Data lakes dos flywheels 288
Estrutura de pastas do data lake 289
Gerenciamento do data lake 290
Políticas do IAM e permissões 291
Configure as permissões de usuário do IAM 291
Configurar permissões para AWS KMS chaves 292
Criar um perfil de acesso a dados 292
Configurar flywheels (console) 292
Crie um flywheel 292
Atualize um flywheel 295
Exclua um flywheel 295
Configurar flywheels (API) 296
Crie um flywheel para um modelo existente 296
Crie um flywheel para um novo modelo 297
Descreva um flywheel. 297
Atualize um flywheel 298
Exclua um flywheel 299
Liste os flywheels 299
Configurar conjuntos de dados 300
Criar um conjunto de dados (console) 300
Criar um conjunto de dados (API) 301
Descrever um conjunto de dados 301
Iterações do flywheel 302
Fluxo de trabalho da iteração 302
Gerenciamento de iterações (console) 303
Gerenciamento de iterações (API) 304
Utilizar flywheels 306
Análise em tempo real 307
Trabalhos assíncronos 307
Gerenciar endpoints 308
Visão geral dos endpoints 308
Utilizar endpoints 309
Monitoramento de endpoints 310
Atualizando endpoints 313

Usando Trusted Advisor 315
Endpoints do Amazon Comprehend subutilizados 315
Risco de acesso ao endpoint do Amazon Comprehend 317
Excluir endpoints 319
Ajuste de escala automático com endpoints 320
Monitoramento do objetivo 321
Escalabilidade programada 326
Tags 330
Marcar um novo recurso 331
Visualizar, editar e excluir tags 332
Exemplos de código 334
Conceitos básicos 335
Ações 336
Cenários 408
Criar uma aplicação de transmissão do Amazon Transcribe 408
Criar um chatbot Amazon Lex 409
Crie um aplicativo de mensagem 410
Criar uma aplicação para analisar o feedback dos clientes 411
Detecte elementos do documento 417
Detectar entidades em texto extraído de uma imagem 423
Execute uma tarefa de modelagem de tópicos em dados de amostra 424
Treinar um classificador personalizado e classificar documentos 429
Segurança 442
Proteção de dados 443
Criptografia do KMS no Amazon Comprehend 444
Prevenção contra o ataque do “substituto confuso” em todos os serviços 447
Utilizar uma nuvem privada virtual (VPC) 449
endpoint da VPC (AWS PrivateLink) 456
Gerenciamento de Identidade e Acesso 458
Público 459
Autenticar com identidades 459
Gerenciar o acesso usando políticas 463
Como o Amazon Comprehend funciona com o IAM 466
Exemplos de políticas baseadas em identidade 473
AWS políticas gerenciadas 486
Solução de problemas 490

Log de chamadas de API do Amazon Comprehend com o AWS CloudTrail 492
Amazon Comprehend informações em CloudTrail 492
Exemplo: entradas de arquivo de log do Amazon Comprehend 495
Validação de conformidade 497
Resiliência 497
Segurança da infraestrutura 498
Diretrizes e cotas 499
Regiões do compatíveis 499
Cotas para modelos integrados 500
Análise em tempo real (síncrona) 500
Análise assíncrona 502
Cotas para modelos personalizados 505
Cotas gerais 505
Cotas para endpoints 505
Classificação de documento 506
Reconhecimento de entidade personalizado 511
Cotas para flywheels 515
Cotas gerais para flywheels 515
Cotas de conjuntos de dados para modelos de classificação personalizados 516
Cotas de conjuntos de dados para modelos personalizados de reconhecimento de entidades 516
Tutoriais 518
Analisando insights de avaliações 518
Pré-requisitos 520
Etapa 1: adicionando documentos ao Amazon S3 522
Etapa 2: (somente para CLI) criando um perfil do IAM 526
Etapa 3: executando trabalhos de análise 530
Etapa 4: preparar a saída 534
Etapa 5: visualizando o resultado 547
Utilizar Pontos de acesso object lambda para PII 553
Controlar o acesso a documentos com PII 554
Editar PIIs em documentos 556
Analisando texto com OpenSearch 558
Referência da API 559
Histórico de documentos 560
AWS Glossário 577

..... dlxxviii

O que é o Amazon Comprehend?

O Amazon Comprehend usa o processamento de linguagem natural (PLN) para extrair insights sobre o conteúdo dos documentos. O serviço desenvolve insights por meio do reconhecimento de entidades, frases importantes, linguagem, sentimentos e outros elementos comuns de um documento. Use o Amazon Comprehend para criar novos produtos com base na compreensão da estrutura dos documentos. Por exemplo, usando o Amazon Comprehend, você pode pesquisar menções de produtos em feeds de redes sociais ou fazer uma varredura em um repositório inteiro de documentos em busca de frases-chave.

Você pode acessar os recursos de análise de documentos do Amazon Comprehend usando o console do Amazon Comprehend ou usando o Amazon Comprehend. APIs Você pode executar análises em tempo real para workloads pequenas ou iniciar trabalhos de análise assíncrona para grandes conjuntos de documentos. Você pode usar os modelos pré-treinados que o Amazon Comprehend fornece ou pode treinar seus próprios modelos personalizados para classificação e reconhecimento de entidades.

O Amazon Comprehend pode armazenar seu conteúdo para melhorar continuamente a qualidade dos modelos pré-treinados. Para saber mais, consulte Perguntas frequentes do Amazon Comprehend.

Todos os recursos do Amazon Comprehend aceitam documentos de texto UTF-8 como entrada. Além disso, a classificação personalizada e o reconhecimento personalizado de entidades aceitam arquivos de imagem, arquivos em PDF e arquivos do Word como entrada.

O Amazon Comprehend pode examinar e analisar documentos em vários idiomas, dependendo do atributo específico. Para obter mais informações, consulte Idiomas compatíveis no Amazon Comprehend. A capacidade Idioma dominante do Amazon Comprehend pode examinar documentos e determinar o idioma dominante para uma seleção muito maior de idiomas.

Tópicos

- Insights do Amazon Comprehend
- Amazon Comprehend Custom
- Flywheels
- Clusters de documentos (modelos de tópicos)
- Exemplos

- Benefícios
- Preços do Amazon Comprehend
- Você é um usuário iniciante do Amazon Comprehend?

Insights do Amazon Comprehend

O Amazon Comprehend usa um modelo pré-treinado para examinar e analisar um documento ou conjunto de documentos para coletar insights sobre ele. Esse modelo é treinado continuamente em um grande corpo de texto para que você não precise fornecer dados de treinamento.

O Amazon Comprehend analisa os seguintes tipos de insights:

- Entidades: referências aos nomes de pessoas, lugares, itens e locais contidos em um documento.
- Frases-chave: frases que aparecem em um documento. Por exemplo, um documento sobre um jogo de basquete pode retornar os nomes das equipes, o nome do local e o score final.
- Informações de identificação pessoal (PII): dados pessoais que podem identificar um indivíduo, como endereço, número de conta bancária ou número de telefone.
- Idioma: o idioma dominante de um documento.
- Sentimento: o sentimento dominante de um documento, que pode ser positivo, neutro, negativo ou misto.
- Sentimento direcionado: os sentimentos associados a entidades específicas em um documento. O sentimento de cada ocorrência de entidade pode ser positivo, negativo ou misto.
- Sintaxe: as partes do discurso de cada palavra no documento.

Para obter mais informações, consulte Insights.

Amazon Comprehend Custom

Você pode personalizar o Amazon Comprehend de acordo com seus requisitos específicos sem o conjunto de habilidades necessário para criar soluções de PNL baseadas em machine learning. Usando o machine learning automático, ou AutoML, o Amazon Comprehend Custom compila modelos de NLP personalizados em seu nome, usando dados que você já possui.

Classificação personalizada: crie modelos de classificação personalizada (classificadores) para organizar seus documentos em suas próprias categorias.

Reconhecimento personalizado de entidades: crie modelos personalizados de reconhecimento de entidades (reconhecedores) que possam analisar o texto de acordo com seus termos específicos e frases baseadas em substantivos.

Para obter mais informações, consulte Amazon Comprehend Custom.

Flywheels

Use flywheels para simplificar o processo de treinamento e gerenciamento de versões personalizadas de modelos ao longo do tempo. Um flywheel ajuda a orquestrar as tarefas associadas ao treinamento e à avaliação de novas versões de um modelo. Flywheels são compatíveis com modelos personalizados de texto simples para classificação personalizada e reconhecimento de entidade personalizada. Para obter mais informações, consulte Flywheels.

Clusters de documentos (modelos de tópicos)

Você também pode usar o Amazon Comprehend para examinar um corpus de documentos para organizá-los com base em palavras-chave semelhantes. O cluster de documentos (modelos de tópicos) é útil para organizar um grande corpus de documentos em tópicos ou clusters semelhantes com base na frequência de palavras. Para obter mais informações, consulte Modelagem de tópicos.

Exemplos

Os exemplos a seguir mostram como usar as operações do Amazon Comprehend em seus aplicativos.

Example 1: Encontre documentos sobre um assunto

Encontre os documentos sobre um assunto específico usando os modelos de tópicos do Amazon Comprehend. Faça uma varredura em um conjunto de documentos para determinar os tópicos discutidos e encontrar os documentos associados a cada tópico. Você pode especificar o número de tópicos que o Amazon Comprehend deve retornar a partir do conjunto de documentos.

Example 2: Descubra como os clientes se sentem em relação aos seus produtos

Se sua empresa publica um catálogo, deixe o Amazon Comprehend dizer o que os clientes pensam dos seus produtos. Envie cada comentário de clientes para a operação `DetectSentiment` e ela dirá se os clientes são positivos, negativos, neutros ou confusos em relação a um produto.

Example 3: Descubra o que é importante para seus clientes

Use os modelos de tópicos do Amazon Comprehend para descobrir os tópicos sobre os quais seus clientes estão falando em seus fóruns e quadros de mensagens e, em seguida, use a detecção de entidades para determinar as pessoas, os lugares e as coisas que eles associam ao tópico. Use a análise de sentimentos para determinar como seus clientes se sentem em relação a um tópico.

Benefícios

Os benefícios do uso do Amazon Comprehend incluem:

- Integre um processamento de linguagem natural potente em seus aplicativos: o Amazon Comprehend elimina a complexidade de compilar capacidades de análise de texto em seus aplicativos, disponibilizando um processamento de linguagem natural potente e preciso com uma API simples. Você não precisa de experiência em análise textual para aproveitar os insights que o Amazon Comprehend produz.
- Processamento de linguagem natural baseado em aprendizado profundo: o Amazon Comprehend usa tecnologia de aprendizado profundo para analisar textos com precisão. Nossos modelos são constantemente treinados com novos dados em vários domínios para melhorar a precisão.
- Escalabilidade de processamento de linguagem natural: o Amazon Comprehend permite que você analise milhões de documentos para que você possa descobrir os insights que eles contêm.
- Integrado com outros AWS serviços — o Amazon Comprehend foi projetado para funcionar perfeitamente com outros serviços, AWS como Amazon S3, e. AWS KMS AWS Lambda Armazene seus documentos no Amazon S3 ou analise dados em tempo real com o Firehose. O suporte para AWS Identity and Access Management (IAM) facilita o controle seguro do acesso às operações do Amazon Comprehend. Ao usar o IAM, você pode criar e gerenciar usuários e grupos para conceder acesso apropriado aos desenvolvedores e usuários finais.
- Criptografia dos resultados de saída e dos dados de volume: o Amazon S3 já permite que você criptografe seus documentos de entrada, e o Amazon Comprehend amplia isso ainda mais. Usando sua própria chave do KMS, você pode criptografar os resultados de saída do seu trabalho e os dados no volume de armazenamento anexado à instância de computação que processa o trabalho de análise. O resultado é uma segurança significativamente aprimorada.
- Baixo custo: com o Amazon Comprehend, não há taxas mínimas nem compromissos antecipados. Você paga pelos documentos que analisa e pelos modelos personalizados que treina.

Preços do Amazon Comprehend

Com o Amazon Comprehend, você paga apenas pelos recursos que usa. Se você for um cliente novo da AWS , poderá começar a usar o Amazon Comprehend gratuitamente. Para obter mais informações, consulte Nível de uso gratuito da AWS.

Há uma taxa de uso para executar trabalhos de análise em tempo real ou assíncronos. Você paga para treinar modelos personalizados e paga pelo gerenciamento de modelos personalizados. Para solicitações em tempo real usando modelos personalizados, você paga pelo endpoint desde o momento em que inicia seu endpoint até excluir o endpoint. Não há cobrança adicional pelo uso de flywheels. No entanto, a execução de uma iteração do flywheel implica em cobranças padrão para treinar uma nova versão do modelo e armazenar os dados dele.

Para obter as tarifas e informações adicionais detalhadas, consulte Preços do Amazon Comprehend.

Você é um usuário iniciante do Amazon Comprehend?

Se você estiver usando o Amazon Comprehend pela primeira vez, recomendamos que leia as seções a seguir nesta ordem:

1. Como funciona: esta seção apresenta os conceitos do Amazon Comprehend.
2. Configuração: nesta seção, você configura uma conta e configura o AWS CLI.
3. Introdução ao Amazon Comprehend: nesta seção, você executa um trabalho de análise do Amazon Comprehend.
4. Tutorial: analisando insights de avaliações de clientes com o Amazon Comprehend: nesta seção, você realiza análises de sentimentos e entidades e visualiza os resultados.
5. Referência de API do Amazon Comprehend: documentação de referência para as operações do Amazon Comprehend.

AWS fornece os seguintes recursos para aprender sobre o serviço Amazon Comprehend:

- O Blog de Machine Learning da AWS inclui artigos úteis sobre o Amazon Comprehend.
- Os Recursos do Amazon Comprehend oferecem vídeos e tutoriais úteis sobre o Amazon Comprehend.

Como funciona

O Amazon Comprehend usa um modelo pré-treinado para coletar insights sobre um documento ou um conjunto de documentos. Esse modelo é treinado continuamente em um grande corpo de texto para que você não precise fornecer dados de treinamento.

Você pode usar o Amazon Comprehend para criar seus próprios modelos personalizados para classificação personalizada e reconhecimento personalizado de entidades. Você pode usar Flywheels para ajudar a gerenciar os modelos personalizados.

O Amazon Comprehend fornece modelagem de tópicos usando um modelo integrado. A modelagem de tópicos examina um corpus de documentos e os organiza com base em palavras-chave semelhantes dentro deles.

O Amazon Comprehend fornece modos de processamento de documentos síncrono e assíncrono. Use o modo síncrono para processar um documento ou um lote de até 25 documentos. Use uma tarefa assíncrona para processar um grande número de documentos.

O Amazon Comprehend trabalha AWS Key Management Service com AWS KMS() para fornecer criptografia aprimorada para seus dados. Para obter mais informações, consulte Criptografia do KMS no Amazon Comprehend.

Principais conceitos

- Insights
- Amazon Comprehend Custom
- Modelagem de tópicos
- Modos de processamento de documentos

Insights

O Amazon Comprehend pode analisar um documento ou conjunto de documentos para coletar informações sobre ele. Alguns dos insights que o Amazon Comprehend desenvolve sobre um documento incluem:

- Entidades: o Amazon Comprehend retorna uma lista de entidades, como pessoas, lugares e locais, identificadas em um documento.

- Eventos: o Amazon Comprehend detecta tipos específicos de eventos e detalhes relacionados.
- Frases chave: o Amazon Comprehend extrai frases-chave que aparecem em um documento. Por exemplo, um documento sobre um jogo de basquete pode retornar os nomes das equipes, o nome do local e o score final.
- Informações de identificação pessoal (PII) : o Amazon Comprehend analisa documentos para detectar dados pessoais que identificam um indivíduo, como endereço, número de conta bancária ou número de telefone.
- Idioma dominante: o Amazon Comprehend identifica a linguagem dominante em um documento. O Amazon Comprehend pode reconhecer 100 idiomas.
- Sentimento: o Amazon Comprehend determina o sentimento dominante de um documento. O sentimento pode ser positivo, neutro, negativo ou misto.
- Sentimento direcionado: o Amazon Comprehend determina o sentimento de entidades específicas mencionadas em um documento. O sentimento de cada menção pode ser positivo, neutro, negativo ou misto.
- Análise de sintaxe: o Amazon Comprehend analisa cada palavra em seu documento e determina a parte da fala da palavra. Por exemplo, na frase “Está chovendo hoje em Seattle”, “isso” é identificado como um pronome, “chovendo” é identificado como um verbo e “Seattle” é identificado como um nome próprio.

Entidades

Uma entidade é uma referência textual ao nome exclusivo de um objeto do mundo real, como pessoas, lugares e itens comerciais, e a referências precisas a medidas, como datas e quantidades.

Por exemplo, no texto “John mudou-se para Mockingbird Lane, 1313 em 2012”, “John” pode ser reconhecido como um `PERSON`, “Mockingbird Lane, 1313” pode ser reconhecido como um `LOCATION` e “2012" pode ser reconhecido como um `DATE`.

Cada entidade também tem uma pontuação que indica o nível de confiança que o Amazon Comprehend tem de que ele detectou corretamente o tipo de entidade. Você pode filtrar as entidades com pontuações mais baixas para reduzir o risco de usar detecções incorretas.

A tabela a seguir lista os tipos de entidades.

Tipo	Descrição
COMMERCIAL_ITEM	Um produto de marca
DATE	Uma data completa (por exemplo, 25/11/2017), dia (terça-feira), mês (maio) ou hora (8h30min)
EVENTO	Um evento, como um festival, concerto, eleição etc.
LOCALIZAÇÃO	Um local específico, como um país, cidade, lago, prédio etc.
ORGANIZATION	Grandes organizações, como governo, empresa, religião, equipe esportiva etc.
OUTRO	Entidades que não se encaixam em nenhuma das outras categorias de entidades.
PERSON	Indivíduos, grupos de pessoas, apelidos, personagens fictícios
QUANTITY	Um valor quantificado, como moedas, porcentagens, números, bytes etc.
TÍTULO	Nome oficial dado a qualquer criação ou trabalho criativo, como filmes, livros, músicas etc.

As operações de detecção de entidades podem ser realizadas usando qualquer uma das linguagens principais compatíveis com o Amazon Comprehend. Isso inclui somente a detecção de entidades predefinida (não personalizada). Todos os documentos devem estar no mesmo idioma.

Você pode usar qualquer uma das operações de API a seguir para detectar entidades em um documento ou conjunto de documentos.

- DetectEntities
- BatchDetectEntities
- StartEntitiesDetectionJob

As operações retornam uma lista de objetos de Entidades de API, uma para cada entidade no documento. A operação `BatchDetectEntities` retorna uma lista de objetos `Entity`, uma lista

para cada documento no lote. A operação `StartEntitiesDetectionJob` inicia uma tarefa assíncrona que produz um arquivo contendo uma lista de objetos `Entity` para cada documento na tarefa.

O exemplo a seguir é a resposta da operação `DetectEntities`.

```
{
    "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"
}
```

Eventos

Use a detecção de eventos a fim de analisar documentos de texto para tipos específicos de eventos e suas entidades relacionadas. O Amazon Comprehend é compatível com a detecção de eventos em grandes coleções de documentos usando tarefas de análise assíncrona. Para mais informações sobre eventos, incluindo exemplos de tarefas de análise de eventos, consulte Anunciando o lançamento do Amazon Comprehend Events

Entidades

Do texto de entrada, o Amazon Comprehend extrai uma lista de entidades relacionadas ao evento detectado. Uma entidade pode ser um objeto do mundo real, como uma pessoa, um lugar ou um local; uma entidade também pode ser um conceito, como uma medida, data ou quantidade. Cada ocorrência de uma entidade é identificada por uma menção, que é uma referência textual à entidade no texto de entrada. Para cada entidade exclusiva, todas as menções são agrupadas em uma lista.

Essa lista fornece detalhes de cada local no texto de entrada em que a entidade ocorre. O Amazon Comprehend detecta somente as entidades associadas aos tipos de eventos compatíveis.

Cada entidade associada a um tipo de evento compatível retorna com os seguintes detalhes relacionados:

- Menções: detalhes de cada ocorrência da mesma entidade no texto de entrada.
  - BeginOffset: um deslocamento de caracteres no texto de entrada que mostra onde a menção começa (o primeiro caractere está na posição 0).
  - EndOffset: um deslocamento de caracteres no texto de entrada que mostra onde a menção termina.
  - Score: o nível de confiança que o Amazon Comprehend tem na precisão do tipo da entidade.
  - GroupScore: o nível de confiança do Amazon Comprehend de que a menção está corretamente agrupada com outras menções da mesma entidade.
  - Texto: o texto da entidade.
  - Tipo: o tipo da entidade. Para todos os tipos de entidades compatíveis, consulte Tipos de entidade.

Eventos

O Amazon Comprehend retorna a lista de eventos (dos tipos de eventos compatíveis) que ele detecta no texto de entrada. Cada evento retorna com os seguintes detalhes relacionados:

- Tipo: o tipo do evento. Para uma lista dos tipos de evento compatíveis, consulte Tipos de eventos.
- Argumentos: uma lista de argumentos relacionados ao evento detectado. Um argumento consiste em uma entidade relacionada ao evento detectado. O papel do argumento descreve o relacionamento, como quem fez o quê, onde e quando.
  - EntityIndex: um valor de índice que identifica uma entidade da lista de entidades que o Amazon Comprehend retornou para essa análise.
  - Perfil: o tipo de argumento, que descreve como a entidade desse argumento está relacionada ao evento. Para todos os tipos de argumentos compatíveis, consulte Tipos de argumento.
  - Score: o nível de confiança que o Amazon Comprehend tem na precisão do tipo da entidade.
- Gatilhos: uma lista de gatilhos para o evento detectado. Um gatilho é uma única palavra ou frase que indica a ocorrência do evento.

- BeginOffset: um deslocamento de caracteres no texto de entrada que mostra onde o gatilho começa (o primeiro caractere está na posição 0).
- EndOffset: um deslocamento de caracteres no texto de entrada que mostra onde o gatilho termina.
- Score: o nível de confiança que o Amazon Comprehend tem na precisão da detecção.
- Texto: o texto do gatilho.
- GroupScore: o nível de confiança do Amazon Comprehend de que o gatilho está corretamente agrupado com outros gatilhos para o mesmo evento.
- Tipo: o tipo de evento que esse gatilho indica.

Detecte o formato de resultados de eventos

Quando sua tarefa de detecção de eventos é concluída, o Amazon Comprehend grava os resultados da análise no local de saída do Amazon S3 especificado ao iniciar a tarefa.

Para cada evento detectado, a saída fornece detalhes no seguinte formato:

```
{
   "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"
        }, ...
      ]
    }, ...
  ]
}
```

Tipos compatíveis para entidades, eventos e argumentos

Tipos de entidade

Tipo	Descrição
DATE	Qualquer referência a uma data ou hora, seja específica ou geral.
FACILITY	Edifícios, aeroportos, rodovias, pontes e outras estruturas permanentes feitas pelo homem e melhorias imobiliárias.
LOCALIZAÇÃO	Localizações físicas, como ruas, cidades, estados, países, corpos d'água ou coordenadas geográficas.
MONETARY_VALUE	O valor de algo nos EUA ou em outra moeda. O valor pode ser específico ou aproximado.
ORGANIZATION	Empresas e outros grupos de pessoas definidos por uma estrutura organizacional estabelecida.

Tipo	Descrição
PERSON	Os nomes ou apelidos de indivíduos ou personagens fictícios.
PERSON_TITLE	Qualquer título que descreva uma pessoa, que geralmente é uma categoria de emprego (como CEO) ou honorífico (como Sr.).
QUANTITY	Um número ou valor e a unidade de medida.
STOCK_CODE	Um símbolo de cotação da bolsa, como AMZN, um International Securities Identific ation Number (ISIN — Número Internaci onal de Identificação de Valores Mobiliários), Committee on Uniform Securities Identification Procedures (CUSIP — Comitê de Procedime ntos Uniformes de Identificação de Valores Mobiliários) ou Stock Exchange Daily Official List (SEDOL — Lista Oficial Diária da Bolsa de Valores).

Tipos de eventos

Tipo	Descrição
BANKRUPTCY	Um processo legal envolvendo uma pessoa ou empresa incapaz de pagar dívidas pendentes.
EMPLOYMENT	Ocorre quando um funcionário é contratado, demitido, aposentado ou muda de outra forma o estado de emprego.
CORPORATE_ACQUISTION	Ocorre quando uma empresa obtém a posse da maioria ou de todas as ações ou ativos físicos de outra empresa para obter o controle dela.

Tipo	Descrição
INVESTMENT_GENERAL	Ocorre quando uma pessoa ou empresa compra um ativo com a perspectiva de gerar renda ou valorização futura.
CORPORATE_MERGER	Ocorre quando duas ou mais empresas se unem para criar uma nova entidade legal.
IPO	Uma initial public offering (IPO — oferta pública inicial) de ações de uma empresa privada ao público em uma nova emissão de ações.
RIGHTS_ISSUE	Um grupo de direitos oferecidos aos acionistas existentes para comprar ações adicionais, conhecidas como garantias de subscrição, na proporção de suas participações existentes.
SECONDARY_OFFERING	Uma oferta de valores mobiliários feita por um acionista de uma empresa.
SHELF_OFFERING	Uma disposição da Securities and Exchange Commission (SEC — Comissão de Valores Mobiliários) que permite que um emissor registre uma nova emissão de título e venda partes da emissão por um período de tempo sem registrar novamente o título ou incorrer em penalidades. Também conhecido como registro de prateleira.
TENDER_OFFERING	Uma oferta para comprar algumas ou todas as ações dos acionistas de uma empresa.
STOCK_SPLIT	Ocorre quando o conselho de administração de uma empresa aumenta o número de ações em circulação emitindo mais ações aos acionistas atuais. Este evento também se aplica a divisões reversas de ações.

Tipos de argumento

Tipos de argumentos para BANKRUPTCY

Tipo de argumento	Descrição
FILER	A pessoa ou empresa que declara a falência.
DATE	A data ou hora da falência.
PLACE	Local ou instalação onde (ou mais próximo de onde) ocorreu a falência.

Tipos de argumentos para EMPLOYMENT

Tipo	Descrição
EMPLOYEES	A pessoa empregada por uma empresa.
EMPLOYEE_TITLE	O título do funcionário.
EMPLOYER	A pessoa ou empresa que emprega o funcionário.
START_DATE	A data ou hora de início do emprego.
END_DATE	A data ou hora de término do emprego.

Tipos de argumentos para CORPORATE_ACQUISTION e INVESTMENT_GENERAL

Tipo	Descrição
AMOUNT	O valor monetário associado à transação.
INVESTEE	A pessoa ou empresa associada ao investimento.
INVESTOR	A pessoa ou empresa que investe no ativo.

Tipo	Descrição
DATE	A data ou a hora da aquisição ou do investimento.
PLACE	Local onde (ou mais próximo de onde) a aquisição ou investimento ocorreu.

Tipos de argumentos para CORPORATE_MERGER

Tipo	Descrição
DATE	A data ou a hora da fusão.
NEW_COMPANY	A nova entidade legal resultante da fusão.
PARTICIPANT	A empresa envolvida na fusão.

Tipos de argumentos para IPO, RIGHTS_ISSUE, SECONDARY_OFFERING, SHELF_OFFERING e TENDER_OFFERING

Tipo	Descrição
EXPIRE_DATE	A data ou hora de expiração da oferta.
INVESTOR	A pessoa ou empresa que investe no ativo.
OFFEREE	A pessoa ou empresa que recebe a oferta.
OFFERING_AMOUNT	O valor monetário associado à oferta.
OFFERING_DATE	A data ou hora da oferta.
OFFEROR	A pessoa ou empresa que está iniciando a oferta.
OFFEROR_TOTAL_VALUE	O valor monetário total associado à oferta.
RECORD_DATE	A data ou hora de registro da oferta.

Tipo	Descrição
SELLING_AGENT	A pessoa ou empresa que facilita a venda da oferta.
SHARE_PRICE	O valor monetário associado ao preço da ação.
SHARE_QUANTITY	O número de ações associadas à oferta.
UNDERWRITERS	A empresa associada à subscrição da oferta.

Tipos de argumentos para STOCK_SPLIT

Tipo	Descrição
COMPANY	A empresa que emite ações do desdobramento de ações.
DATE	A data ou hora do desdobramento de ações.
SPLIT_RATIO	A proporção entre o aumento do novo número de ações em circulação e o número atual de ações antes de seu desdobramento.

Frases chave

Uma frase-chave é uma string contendo uma frase nominal que descreve uma coisa específica. Geralmente consiste em um substantivo e nos modificadores que o distinguem. Por exemplo, “dia” é um substantivo; “um lindo dia” é uma frase nominal que inclui um artigo (“a”) e um adjetivo (“lindo”). Cada frase-chave inclui um score que indica o nível de confiança que o Amazon Comprehend tem de que a string é uma frase nominal. Use o score para determinar se a detecção tem confiança alta o suficiente para sua aplicação.

As operações de detecção de frases-chave podem ser realizadas usando qualquer uma das linguagens principais compatíveis com o Amazon Comprehend. Todos os documentos devem estar no mesmo idioma.

Use qualquer uma das operações a seguir para detectar frases-chave em um documento ou conjunto de documentos.

- DetectKeyPhrases
- BatchDetectKeyPhrases
- StartKeyPhrasesDetectionJob

As operações retornam uma lista de KeyPhraseobjetos, uma para cada frase-chave no documento. A operação `BatchDetectKeyPhrases` retorna uma lista de objetos `KeyPhrase`, uma para cada documento no lote. A operação `StartKeyPhrasesDetectionJob` inicia uma tarefa assíncrona que produz um arquivo contendo uma lista de objetos `KeyPhrase` para cada documento na tarefa.

O exemplo a seguir é a resposta da operação `DetectKeyPhrases`.

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

Idioma dominante

Você pode usar o Amazon Comprehend para examinar o texto e determinar o idioma dominante. O Amazon Comprehend identifica o idioma usando identificadores segundo o RFC 5646: se houver um identificador ISO 639-1 de 2 letras, com uma subtag regional, se necessário, ele usará essas informações. Caso contrário, ele usará o código ISO 639-2 de 3 letras.

Para obter mais informações sobre o RFC 5646, consulte Tags para identificação de idiomas no site IETF Tools.

A resposta inclui uma pontuação que indica o nível de confiança que o Amazon Comprehend tem de que um determinado idioma é o idioma dominante no documento. Cada pontuação é independente das outras pontuações. A pontuação não indica que um idioma compõe uma porcentagem específica de um documento.

Se um documento longo (como um livro) contiver vários idiomas, você poderá dividi-lo em partes menores e executar a operação `DetectDominantLanguage` nas partes individuais. Em seguida, você poderá agregar os resultados para determinar a porcentagem de cada idioma no documento mais longo.

A detecção de linguagens do Amazon Comprehend tem as seguintes limitações:

- Ela não oferece suporte à detecção de linguagem fonética. Por exemplo, ele não detecta “arigato” como japonês ou “nihao” como chinês.
- O recurso pode ter dificuldade em distinguir pares de idiomas próximos, como indonésio e malaio; ou bósnio, croata e sérvio.
- Para obter melhores resultados, forneça pelo menos 20 caracteres de texto de entrada.

O Amazon Comprehend detecta os seguintes idiomas.

Código	Idioma
af	Africâner
am	Amárico
ar	Árabe
as	Assamês
az	Azerbaijano
ba	Bashkir
be	Bielorrusso

Código	Idioma
bn	Bengali
bs	Bósnio
bg	Búlgaro
ca	Catalão
ceb	Cebuano
cs	Tcheco
cv	Tchuvache
cy	Galês
da	Dinamarquês
de	Alemão
el	Grego
en	Inglês
eo	Esperanto
et	Estoniano
eu	Basco
fa	Persa
fi	Finlandês
fr	Francês
gd	Gaélico escocês
ga	Irlandês

Código	Idioma
gl	Galego
gu	Gujarati
ht	Haitiano
he	Hebraico
ha	Hauçá
hi	Hindi
hr	Croata
hu	Húngaro
hy	Armênio
ilo	Ilocano
id	Indonésio
is	Islandês
it	Italiano
jv	Javanês
ja	Japonês
kn	Canarês
ka	Georgiano
kk	Cazaque
km	Khmer Central
ky	Quirguiz

Código	Idioma
ko	Coreano
ku	Curdo
lo	Laosiano
la	Latim
lv	Letão
lt	Lituano
lb	Luxemburguês
ml	Malaiala
mt	Maltês
mr	Marati
mk	Macedônio
mg	Malgaxe
mn	Mongol
ms	Malaio
my	Birmanês
ne	Nepalês
new	Neuari
nl	Holandês
no	Norueguês
or	Oriá

Código	Idioma
om	Oromo
pa	Punjabi
pl	Polonês
pt	Português
ps	Pastó
qu	Quechua
ro	Romeno
ru	Russo
sa	Sânscrito
si	Cingalês
sk	Eslovaco
sl	Esloveno
sd	Sindi
so	Somali
es	Espanhol
sq	Albanês
sr	Sérvio
su	Sudanês
sw	Suaíli
sv	Sueco

Código	Idioma
ta	Tâmil
tt	Tatárico
te	Telugo
tg	Tadjique
tl	Tagalo
th	Tailandês
tk	Turcomano
tr	Turco
ug	Uigur
uk	Ucraniano
ur	Urdu
uz	Uzbeque
vi	Vietnamita
yi	Iídiche
yo	Iorubá
zh	Chinês (simplificado)
zh-TW	Chinês (tradicional)

Você pode usar qualquer uma das operações a seguir para detectar o idioma dominante em um documento ou conjunto de documentos.

- DetectDominantLanguage
- BatchDetectDominantLanguage

- StartDominantLanguageDetectionJob

A `DetectDominantLanguage` operação retorna um DominantLanguageobjeto. A operação `BatchDetectDominantLanguage` retorna uma lista de objetos `DominantLanguage`, uma para cada documento no lote. A operação `StartDominantLanguageDetectionJob` inicia um trabalho assíncrono que produz um arquivo contendo uma lista de objetos `DominantLanguage`, uma para cada documento no trabalho.

O exemplo a seguir é a resposta da operação `DetectDominantLanguage`.

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

Sentimento

Use o Amazon Comprehend para determinar o sentimento do conteúdo em documentos de texto codificados em UTF-8. Por exemplo, você pode usar a análise de sentimentos para determinar os sentimentos dos comentários em uma postagem no blog e determinar se seus leitores gostaram da postagem.

Você pode determinar a opinião sobre documentos em qualquer um dos principais idiomas suportados pelo Amazon Comprehend. Todos os documentos em um trabalho devem estar no mesmo idioma.

A determinação do sentimento retorna os seguintes valores:

- Positivo: o texto expressa um sentimento geral positivo.
- Negativo: o texto expressa um sentimento geral negativo.
- Misto: o texto expressa sentimentos positivos e negativos.
- Neutro: o texto não expressa sentimentos positivos ou negativos.

Você pode usar qualquer uma das operações de API a seguir para detectar o sentimento de um documento ou de um conjunto de documentos.

- DetectSentiment
- BatchDetectSentiment
- StartSentimentDetectionJob

As operações retornam o sentimento mais provável para o texto e as pontuações para cada um dos sentimentos. A pontuação representa a probabilidade de o sentimento ter sido detectado corretamente. Por exemplo, no exemplo abaixo, há uma probabilidade de 95% de que o texto tenha um sentimento `Positive`. Há menos de 1% de probabilidade de que o texto tenha um sentimento `Negative`. Você pode usar `SentimentScore` para determinar se a precisão da detecção atende às necessidades da sua aplicação.

A `DetectSentiment` operação retorna um objeto que contém o sentimento detectado e um SentimentScoreobjeto. A operação `BatchDetectSentiment` retorna uma lista de sentimentos e objetos `SentimentScore`, uma para cada documento no lote. A operação `StartSentimentDetectionJob` inicia um trabalho assíncrono que produz um arquivo contendo uma lista de sentimentos e objetos `SentimentScore`, um para cada documento no trabalho.

O exemplo a seguir é a resposta da operação `DetectSentiment`.

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

Sentimento direcionado

O sentimento direcionado fornece uma compreensão granular dos sentimentos associados a entidades específicas (como marcas ou produtos) em seus documentos de entrada.

A diferença entre sentimento direcionado e sentimento é o nível de granularidade nos dados de saída. A análise de sentimentos determina o sentimento dominante para cada documento de entrada, mas não fornece dados para análise posterior. A análise de sentimento direcionado determina o sentimento por entidade para entidades específicas em cada documento de entrada. Você pode analisar os dados de saída para determinar os produtos e serviços específicos que recebem feedback positivo ou negativo.

Por exemplo, em um conjunto de avaliações de restaurantes, um cliente fornece a seguinte avaliação: “Os tacos estavam deliciosos e a equipe foi simpática”. A análise dessa revisão produzirá os seguintes resultados:

- A análise de sentimento determina se o sentimento geral de cada avaliação de restaurante é positivo, negativo, neutro ou misto. Nesse exemplo, o sentimento geral é positivo.
- A análise de sentimento direcionado determina o sentimento por entidades e atributos do restaurante que os clientes mencionam nas avaliações. Nesse exemplo, o cliente fez comentários positivos sobre “tacos” e “equipe”.

O sentimento direcionado fornece os seguintes resultados para cada trabalho de análise:

- Identidade das entidades mencionadas nos documentos.
- Classificação do tipo de entidade para cada menção de entidade.
- O sentimento e uma pontuação de sentimento para cada menção de entidade.
- Grupos de menções (grupos de correferência) que correspondem a uma única entidade.

Você pode usar o console ou a API para executar uma análise de sentimento direcionado. O console e a API oferecem suporte à análise em tempo real e análise assíncrona para sentimento direcionado.

O Amazon Comprehend suporta sentimentos direcionados para documentos no idioma inglês.

Para obter informações adicionais sobre o sentimento direcionado, incluindo um tutorial, consulte Extrair sentimento granular em texto com o Amazon Comprehend Targeted Sentiment no blog de aprendizado de máquina. AWS

Tópicos

- Tipos de entidade
- Grupo de correferência

- Organização do arquivo de saída
- Análise em tempo real usando o console
- Exemplo de saída de sentimento direcionado

Tipos de entidade

O sentimento direcionado identifica os seguintes tipos de entidade. Ele atribui o tipo de entidade OUTRO se a entidade não pertencer a nenhuma outra categoria. Cada menção de entidade no arquivo de saída inclui o tipo de entidade, como `"Type": "PERSON"`.

Definições do tipo de entidade

Tipo de entidade	Definição
PERSON	Os exemplos incluem indivíduos, grupos de pessoas, apelidos, personagens fictícios e nomes de animais.
LOCALIZAÇÃO	Localizações geográficas, como países, cidades, estados, endereços, formações geológicas, corpos d'água, marcos naturais e localizações astronômicas.
ORGANIZATION	Os exemplos incluem governos, empresas, equipes esportivas e religiões.
FACILITY	Edifícios, aeroportos, rodovias, pontes e outras estruturas permanentes feitas pelo homem e melhorias imobiliárias.
MARCA	Organização, grupo ou produtor de um item comercial específico ou linha de produtos.
COMMERCIAL_ITEM	Qualquer item não genérico comprável ou adquirível, incluindo veículos e produtos grandes que tiveram apenas um item produzido.
FILME	Um filme ou programa de televisão. A entidade pode ser o nome completo, um apelido ou um subtítulo.
MÚSICA	Uma música, completa ou parcial. Além disso, coleções de criações musicais individuais, como um álbum ou uma antologia.
LIVRO	Um livro publicado profissionalmente ou autopublicado.

Tipo de entidade	Definição
SOFTWARE	Um produto de software lançado oficialmente.
JOGO	Um jogo, como videogames, jogos de tabuleiro, jogos comuns ou esportes.
TÍTULO_PESSOAL	Títulos oficiais e honoríficos, como Presidente, PhD ou Dr.
EVENTO	Os exemplos incluem festivais, apresentações musicais, eleições, guerras, conferências e eventos promocionais.
DATA	Qualquer referência a uma data ou hora, seja específica ou geral, absoluta ou relativa.
QUANTITY	Todas as medidas junto com suas unidades (moeda, porcentagem, número, bytes etc.).
ATRIBUTO	Um atributo, característica ou traço de uma entidade, como a “qualidade” de um produto, o “preço” de um telefone ou a “velocidade” de uma CPU.
OUTRO	Entidades que não pertencem a nenhuma das outras categorias.

Grupo de correferência

O sentimento direcionado identifica grupos de correferência em cada documento de entrada. Um grupo de correferência é um grupo de menções em um documento que corresponde a uma entidade do mundo real.

Example

No seguinte exemplo de uma avaliação de cliente, “spa” é a entidade com o tipo de entidade `FACILITY`. A entidade tem duas menções adicionais como pronome (“it”).

Organização do arquivo de saída

O trabalho de análise de sentimento direcionado cria um arquivo JSON de saída de texto. O arquivo contém um objeto JSON para cada um dos documentos de entrada. Cada objeto JSON contém os seguintes campos:

- Entidades: uma matriz de entidades encontradas no documento.
- Arquivo: o nome do arquivo do documento de entrada.
- Linha: se o arquivo de entrada for um documento por linha, Entidades conterá o número da linha do documento no arquivo.

> **Note**
>
> Se o sentimento direcionado não identificar nenhuma entidade no texto de entrada, ele retornará uma matriz vazia como resultado de Entidades.

O exemplo a seguir mostra as Entidades para um arquivo de entrada com três linhas de entrada. O formato de entrada é UM_DOC_POR_LINHA, então cada linha de entrada é um documento.

```
{ "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
}
```

Uma entidade na matriz Entidades inclui um agrupamento lógico (chamado de grupo de correferência) das menções de entidade detectadas no documento. Cada entidade tem a seguinte estrutura geral:

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

Uma entidade contém os seguintes campos:

- Menções: uma matriz de menções da entidade no documento. A matriz representa um grupo de correferência. Consulte the section called “Grupo de correferência” para ver um exemplo. A ordem das menções na matriz Mentions é a ordem da respectiva localização (deslocamento) no documento. Cada menção inclui a pontuação do sentimento e a pontuação do grupo dessa menção. A pontuação do grupo indica o nível de confiança do pertencimento dessas menções à mesma entidade.
- DescriptiveMentionIndex— Um ou mais índices na matriz Menções que fornecem o melhor nome para o grupo de entidades. Por exemplo, uma entidade pode ter três menções com os valores de Texto “ABC Hotel”, “ABC Hotel” e “it”. O melhor nome é “ABC Hotel”, que tem um DescriptiveMentionIndex valor de [0,1].

Cada menção inclui os seguintes campos

- BeginOffset— O deslocamento para o texto do documento em que a menção começa.
- EndOffset— O deslocamento para o texto do documento onde a menção termina.
- GroupScore— A confiança de que todas as entidades mencionadas no grupo se relacionam com a mesma entidade.
- Texto: o texto no documento que identifica a entidade.
- Tipo: o tipo de identidade. O Amazon Comprehend oferece suporte a uma variedade de tipos de entidades.
- Pontuação: confiança do modelo de que a entidade é relevante. O intervalo de valores é de zero a um, com um indicando a maior confiança.
- MentionSentiment— Contém o sentimento e a pontuação de sentimento da menção.
- Sentimento: o sentimento da menção. Os valores incluem: POSITIVO, NEUTRO, NEGATIVO e MISTO.
- SentimentScore— Fornece confiança ao modelo para cada um dos sentimentos possíveis. O intervalo de valores é de zero a um, com um indicando a maior confiança.

Os valores de Sentimento têm o seguinte significado:

- Positivo: a menção da entidade expressa um sentimento positivo.
- Negativo: a menção da entidade expressa um sentimento negativo.
- Misto: a menção da entidade expressa sentimentos positivos e negativos.
- Neutro: a menção da entidade não expressa sentimentos positivos nem negativos.

No exemplo a seguir, uma entidade tem somente uma menção no documento de entrada, então DescriptiveMentionIndex é zero (a primeira menção na matriz Menções). A entidade identificada é uma PESSOA com o nome “I”. A pontuação do sentimento é neutra.


```
{"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
}
```

Análise em tempo real usando o console

Você pode usar o console do Amazon Comprehend para executar the section called “Sentimento direcionado” em tempo real. Use o texto de amostra ou cole seu próprio texto na caixa de texto de entrada e escolha Analisar.

No painel Insights, o console exibe três visualizações da análise de sentimento direcionada:

- Texto analisado: exibe o texto analisado e sublinha cada entidade. A cor do sublinhado indica o valor do sentimento (positivo, neutro, negativo ou misto) que a análise atribuiu à entidade. O console exibe mapeamentos de cores no canto superior direito da caixa do texto analisado. Se você passar o cursor sobre uma entidade, o console exibirá um painel pop-up contendo valores de análise (tipo de entidade, pontuação de sentimento) da entidade.
- Resultados: exibe uma tabela contendo uma linha para cada menção de entidade identificada no texto. Para cada entidade, a tabela mostra a entidade e a pontuação da entidade. A linha também inclui o sentimento principal e a pontuação de cada valor de sentimento. Se houver várias menções da mesma entidade, conhecida como the section called “Grupo de correferência”, a tabela exibirá essas menções como um conjunto flexível de linhas associadas à entidade principal.

  Se você passar o mouse sobre uma linha da entidade na tabela Resultados, o console destacará a menção da entidade no painel Texto analisado.

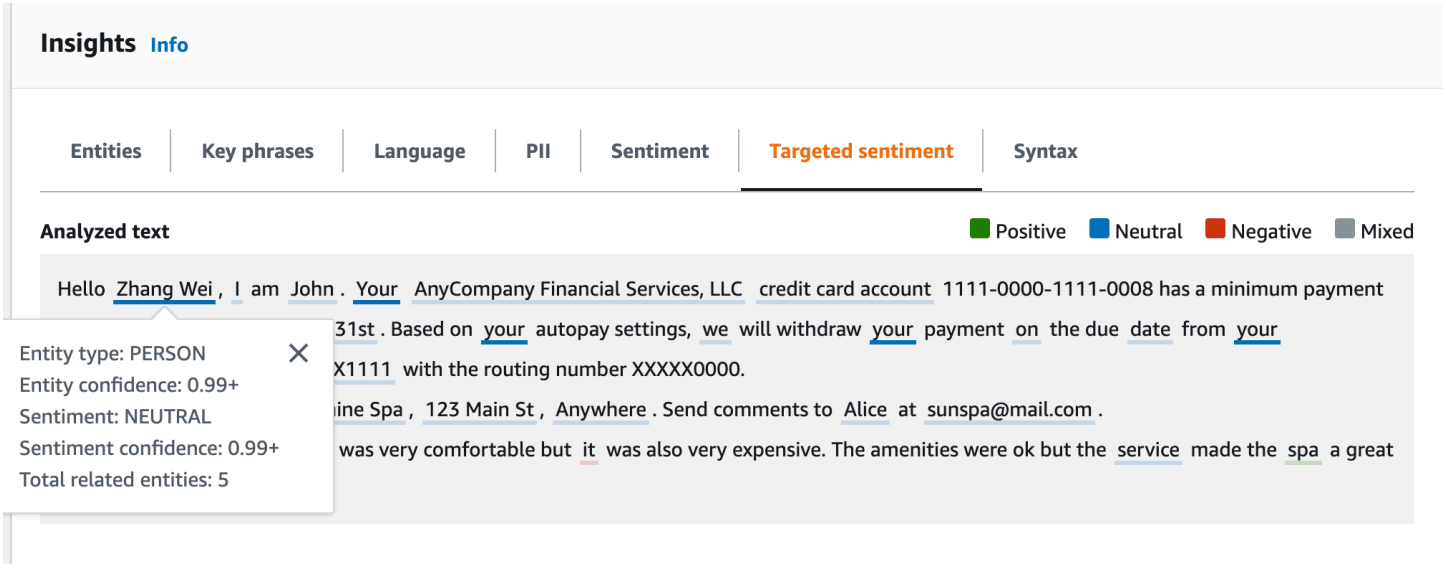
- Integração de aplicação: exibe os valores dos parâmetros da solicitação de API e a estrutura do objeto JSON retornado na resposta da API. Para obter uma descrição dos campos no objeto JSON, consulte the section called “Organização do arquivo de saída”.

Exemplo de análise em tempo real no console

Este exemplo usa o texto a seguir como entrada, que é o texto de entrada padrão fornecido pelo 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.
```

O painel Texto analisado mostra a seguinte saída para esse exemplo. Passe o mouse sobre o texto `Zhang Wei` para ver o painel pop-up dessa entidade.



A tabela Resultados fornece detalhes adicionais sobre cada entidade, incluindo a pontuação da entidade, o sentimento principal e a pontuação de cada sentimento.

▼ 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	-	-

Em nosso exemplo, a análise de sentimento direcionado reconhece que cada menção de seu no texto de entrada é uma referência à entidade pessoal Zhang Wei. O console exibe essas menções como um conjunto de linhas recolhíveis associadas à entidade principal.

▼ 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

O painel de integração de aplicativos exibe o objeto JSON que a DetectTargetedSentiment API gera. Veja a próxima seção para um exemplo detalhado.

Exemplo de saída de sentimento direcionado

O exemplo a seguir mostra o arquivo de saída de um trabalho de análise de sentimento direcionado. O arquivo de entrada consiste em três documentos 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.
```

A análise de sentimento direcionado desse arquivo de entrada produz a seguinte saída.


```
 {"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
}
  }
```

Análise de sintaxe

Use a análise de sintaxe para analisar as palavras do documento e retornar a parte do discurso, ou função sintática, para cada palavra no documento. É possível identificar os substantivos, verbos, adjetivos e assim por diante em seu documento. Use essas informações para uma compreensão mais aprofundada do conteúdo de seus documentos e para entender a relação das palavras no documento.

Por exemplo, você pode procurar os substantivos em um documento e depois procurar os verbos relacionados a esses substantivos. Em uma frase como “Minha avó mudou o sofá”, é possível ver os substantivos “avó” e “sofá” e o verbo “mudou”. Use essas informações para criar aplicativos a fim de analisar texto para combinações de palavras nas quais você está interessado.

Para iniciar a análise, o Amazon Comprehend analisa o texto fonte para encontrar as palavras individuais no texto. Depois que o texto é analisado, cada palavra recebe a parte do discurso que ela absorve no texto de origem.

O Amazon Comprehend pode identificar as seguintes partes do discurso.

Token	Parte do discurso
ADJ	Adjetivo Palavras que normalmente modificam substantivos.

Token	Parte do discurso
ADP	Aposição O núcleo de uma frase preposicional ou pós-pospo sicional.
ADV	Advérbio Palavras que normalmen te modificam verbos. Eles também podem modificar adjetivos e outros advérbios.
AUX	Auxiliar Palavras funcionais que acompanham o verbo de uma frase verbal.
CCONJ	Conjunção coordenadora Uma conjunção coordenativa conecta palavras, frases ou cláusulas em uma frase sem subordinar uma à outra.
CONJ	Conjunção Uma conjunção conecta palavras, frases ou cláusulas em uma frase.
DET	Determinante Artigos e outras palavras que especificam um sintagma nominal específico.

Token	Parte do discurso
INTJ	Interjeição Palavras usadas como exclamação ou parte de uma exclamação.
NOUN	Substantivo Palavras que especificam uma pessoa, lugar, coisa, animal ou ideia.
NUM	Numeral Palavras, normalmente determinantes, adjetivos ou pronomes, que expressam um número.
O	Outros Palavras que não podem ser atribuídas a uma parte da categoria de fala.
PART	Partícula Palavras funcionais associadas a outra palavra ou frase para dar significado.
PRON	Pronome Palavras que substituem substantivos ou frases nominais.

Token	Parte do discurso
PROPN	Substantivo próprio Um substantivo que é o nome de um indivíduo, lugar ou objeto específico.
PUNCT	Pontuação Caracteres não alfabéticos que delimitam o texto.
SCONJ	Conjunção subordinativa Uma conjunção que une uma cláusula dependente a uma frase. Um exemplo de conjunção subordinada é “porque”.
SYM	Símbolo Entidades semelhantes a palavras, como o cifrão ($) ou símbolos matemáticos.
VERB	Verb Palavras que sinalizam eventos e ações.

Para mais informações sobre as partes do discurso, consulte as Tags universais POS no site Universal Dependencies.

As operações retornam tokens que identificam a palavra e a parte do discurso que a palavra representa no texto. Cada token representa uma palavra no texto fonte. Ele fornece a localização da palavra na fonte, a parte do discurso que a palavra assume no texto, a confiança que o Amazon

Comprehend tem de que a parte da fala foi identificada corretamente e a palavra que foi analisada a partir do texto de origem.

A seguir está a estrutura da lista de tokens de sintaxe. Um token de sintaxe é gerado para cada palavra no documento.

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

Cada token fornece as seguintes informações:

- `BeginOffset` e `EndOffset`: fornece a localização da palavra no texto de entrada.
- `PartOfSpeech`: fornece duas informações, a `Tag` que identifica a parte da fala e a `Score` que representa a confiança que a Amazon Comprehend Syntax tem de que a parte da fala foi identificada corretamente.
- `Text`: fornece a palavra que foi identificada.
- `TokenId`: fornece um identificador para o token. O identificador é a posição do token na lista de tokens.

Amazon Comprehend Custom

Você pode personalizar o Amazon Comprehend de acordo com seus requisitos específicos sem o conjunto de habilidades necessário para criar soluções de PNL baseadas em machine learning. Usando o machine learning automático, ou AutoML, o Comprehend Custom cria modelos de PNL personalizados em seu nome, usando dados de treinamento fornecidos.

Processamento de entrada de documentos: o Amazon Comprehend é compatível com o processamento de documentos em uma etapa para classificação personalizada e reconhecimento personalizado de entidades. Por exemplo, você pode inserir uma combinação de documentos de texto simples e documentos semiestruturados (como documentos em PDF, documentos do Microsoft Word e imagens) em um trabalho de análise personalizada. Para obter mais informações, consulte Processamento de documentos.

Classificação personalizada: crie modelos de classificação personalizada (classificadores) para organizar seus documentos em suas próprias categorias. Para cada rótulo de classificação, forneça um conjunto de documentos que melhor represente esse rótulo e treine seu classificador sobre ela. Uma vez treinado, um classificador pode ser usado em qualquer número de conjuntos de documentos sem rótulos. Você pode usar o console para uma experiência sem código ou instalar o SDK da AWS mais recente. Para obter mais informações, consulte Classificação personalizada.

Reconhecimento personalizado de entidades: crie modelos personalizados de reconhecimento de entidades (reconhecedores) que possam analisar o texto de acordo com seus termos específicos e frases baseadas em substantivos. Treine os reconhecedores para extrair termos como números de apólices ou frases que impliquem uma escalada de clientes. Para treinar o modelo, forneça uma lista das entidades e um conjunto de documentos que as contêm. Depois que o modelo é treinado, é possível enviar tarefas de análise para extrair suas entidades personalizadas. Para obter mais informações, consulte Reconhecimento de entidade personalizado.

Modelagem de tópicos

Você pode usar o Amazon Comprehend para examinar o conteúdo de uma coleção de documentos e determinar temas comuns. Por exemplo, você pode fornecer ao Amazon Comprehend uma coleção de artigos de notícias e ele determinará os assuntos, como esportes, política ou entretenimento. O texto nos documentos não precisa estar anotado.

O Amazon Comprehend usa um modelo de aprendizado baseado na alocação latente de Dirichlet para determinar os tópicos em um conjunto de documentos. Ele examina cada documento para determinar o contexto e o significado de uma palavra. O conjunto de palavras que pertencem frequentemente ao mesmo contexto em todo o conjunto do documento compõe um tópico.

Uma palavra é associada a um tópico em um documento com base na prevalência desse tópico em um documento e na afinidade que o tópico tem com a palavra. A mesma palavra pode ser associada a tópicos diferentes em documentos diferentes com base na distribuição de tópicos em um documento específico.

Por exemplo, a palavra “glicose” em um artigo que fala predominantemente sobre esportes pode ser atribuída ao tópico “esportes”, enquanto a mesma palavra em um artigo sobre “medicina” será atribuída ao tópico “medicina”.

Cada palavra associada a um tópico receberá um peso que indica o quanto a palavra ajuda a definir o tópico. O peso é uma indicação de quantas vezes a palavra ocorre no tópico em comparação com outras palavras no tópico, em todo o conjunto de documentos.

Para obter resultados mais precisos, você deve fornecer ao Amazon Comprehend o maior corpo possível com o qual trabalhar. Para obter os melhores resultados:

- Você deve usar pelo menos 1.000 documentos em cada trabalho de modelagem de tópico.
- Cada documento deve ter pelo menos 3 frases.
- Se um documento consistir principalmente em dados numéricos, você deverá removê-lo do corpo.

A modelagem de tópicos é um processo assíncrono. Você envia sua lista de documentos para o Amazon Comprehend de um bucket do Amazon S3 usando a operação. StartTopicsDetectionJob A resposta é enviada para um bucket do Amazon S3. Você pode configurar os buckets de entrada e saída. Obtenha uma lista dos trabalhos de modelagem de tópicos que você enviou usando a ListTopicsDetectionJobsoperação e visualize informações sobre um trabalho usando a DescribeTopicsDetectionJoboperação. O conteúdo entregue aos buckets do Amazon S3 pode conter conteúdo do cliente. Para obter mais informações sobre a remoção de dados confidenciais, consulte Como faço para esvaziar um bucket do S3? ou Como faço para excluir um bucket do S3?.

Os documentos devem ser arquivos de texto no formato UTF-8. Você pode enviar seus documentos de duas maneiras. A tabela a seguir mostra as opções.

Formato	Descrição
Um documento por arquivo	Cada arquivo contém um documento de entrada. Essa opção é melhor para coleções de documentos grandes.
Um documento por linha	A entrada é um único arquivo. Cada linha no arquivo é considerada um documento. Essa opção é melhor para documentos curtos, como publicações em mídias sociais.

Formato	Descrição
	Cada linha deve terminar com uma alimentação de linha (LF, \n), um retorno de carro (CR, \r) ou ambos (CRLF, \r\n). Não é possível usar o separador de linha Unicode (u+2028) para finalizar uma linha.

Para obter mais informações, consulte o tipo de dados InputDataConfig.

Depois que o Amazon Comprehend processar sua coleção de documentos, ele retornará um arquivo compactado contendo dois arquivos, `topic-terms.csv` e `doc-topics.csv`. Para obter mais informações sobre o arquivo de saída, consulte OutputDataConfig.

O primeiro arquivo de saída, `topic-terms.csv`, é uma lista de tópicos na coleção. Para cada tópico, a lista inclui por padrão os principais termos por tópico de acordo com seu peso. Por exemplo, se você der ao Amazon Comprehend uma coleção de artigos de jornal, ele poderá retornar o seguinte para descrever os dois primeiros tópicos da coleção:

Tópico	Prazo	Weight
000	time	0,18533
000	jogo	0,106072
000	player	0,031625
000	temporada	0.023633
000	jogar	0,021118
000	campo	0,024454
000	treinador	0.016012
000	jogos	0.016191
000	futebol	0,015049
000	quarto zagueiro	0,014239

Tópico	Prazo	Weight
001	copo	0,205236
001	alimento	0.040686
001	minutos	0,036062
001	adicionar	0.029697
001	colher de sopa	0,028789
001	óleo	0,021254
001	pimenta	0.022205
001	colher de chá	0,020040
001	vinho	0.016588
001	açúcar	0,015101

Os pesos representam uma distribuição da probabilidade em relação às palavras em um determinado tópico. Como o Amazon Comprehend retorna somente as 10 palavras principais de cada tópico, os pesos não somarão 1,0. Nos raros casos em que houver menos de 10 palavras em um tópico, os pesos somarão 1,0.

As palavras são classificadas por seu poder discriminativo, observando sua ocorrência em todos os tópicos. Normalmente, isso é o mesmo que seu peso, mas em alguns casos, como as palavras “jogar” e “campo” na tabela, isso resulta em uma ordem que não é igual ao peso.

Você pode especificar o número de tópicos que o sistema deve retornar. Por exemplo, se você pedir ao Amazon Comprehend que retorne 25 tópicos, ele retornará os 25 tópicos mais proeminentes da coleção. O Amazon Comprehend pode detectar até 100 tópicos em uma coleção. Escolha o número de tópicos com base no seu conhecimento do domínio. Talvez seja necessário experimentar um pouco para chegar ao número correto.

O segundo arquivo, `doc-topics.csv`, lista os documentos associados a um tópico e a proporção do documento relacionada ao tópico. Se você tiver especificado `ONE_DOC_PER_FILE`, o documento será identificado pelo nome do arquivo. Se você tiver especificado `ONE_DOC_PER_LINE`, o

documento será identificado pelo nome do arquivo e pelo número da linha indexada em 0 dentro do arquivo. Por exemplo, o Amazon Comprehend pode retornar o seguinte para uma coleção de documentos enviados com um documento por arquivo:

Documento	Tópico	Proporção
doc-exemplo1	000	0,999330137
doc-exemplo2	000	0,998532187
doc-exemplo3	000	0,998384574
...		
doc-exemploN	000	3,57E-04

O Amazon Comprehend utiliza informações do conjunto de dados Lemmatization Lists da MBM, que é disponibilizado aqui sob a licença Open Database (L) v1.0. ODb

Modos de processamento de documentos

O Amazon Comprehend oferece suporte a três modos de processamento de documentos. Sua escolha de modo depende do número de documentos que você precisa processar e da rapidez com que você precisa visualizar os resultados:

- Síncrono de documento único: você chama o Amazon Comprehend com um único documento e recebe uma resposta síncrona, imediatamente entregue à sua aplicação (ou ao console).
- Síncrono de vários documentos: você chama a API do Amazon Comprehend com uma coleção de até 25 documentos e recebe uma resposta síncrona.
- Lote assíncrono: para uma coleção grande de documentos, coloque os documentos em um bucket do Amazon S3 e inicie um trabalho assíncrono (usando operações de console ou API) para analisar os documentos. O Amazon Comprehend armazena os resultados da análise no bucket/pasta do S3 que você especifica na solicitação.

Tópicos

- Processamento de documento único
- Processamento síncrono de vários documentos

- Processamento em lote assíncrono

Processamento de documento único

As operações de documento único são operações síncronas que retornam os resultados da análise do documento diretamente para sua aplicação. Use operações síncronas de documento único ao criar uma aplicação interativa que funcione em um documento por vez.

Para obter mais informações sobre as operações síncronas de API, consulte Análise em tempo real usando os modelos integrados (para console) e Análise em tempo real usando a API.

Processamento síncrono de vários documentos

Quando tiver vários documentos que deseja processar, você poderá usar as operações de API `Batch*` para enviar mais de um documento ao Amazon Comprehend por vez. Você pode enviar até 25 documentos em cada solicitação. O Amazon Comprehend envia de volta uma lista de respostas, uma para cada documento na solicitação. As solicitações feitas com essas operações são síncronas. Seu aplicativo chama a operação e, em seguida, aguarda a resposta do serviço.

Usar as `Batch*` operações é idêntico a chamar o documento único APIs para cada um dos documentos na solicitação. O uso deles APIs pode resultar em melhor desempenho para seus aplicativos.

A entrada para cada um deles APIs é uma estrutura JSON contendo os documentos a serem processados. Para todas as operações, exceto `BatchDetectDominantLanguage`, você deve definir o idioma de entrada. Você pode definir apenas um idioma de entrada para cada solicitação. Por exemplo, veja a seguir a entrada para a operação `BatchDetectEntities`. Ela contém dois documentos e está em inglês.

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

A resposta de uma operação `Batch*` contém duas listas, `ResultList` e `ErrorList`. `ResultList` contém um registro para cada documento que foi processado com sucesso. O resultado de

cada documento na solicitação é idêntico ao resultado que você obteria se executasse uma única operação de documento no documento. Os resultados de cada documento recebem um índice com base na ordem dos documentos no arquivo de entrada. A resposta da operação `BatchDetectEntities` é:

```
{
   "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": []
}
```

Quando ocorre um erro na solicitação, a resposta contém um `ErrorList` que identifica os documentos que continham um erro. O documento é identificado por seu índice na lista de entrada. Por exemplo, a entrada a seguir para a operação `BatchDetectLanguage` contém um documento que não pode ser processado:

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

A resposta do Amazon Comprehend inclui uma lista de erros que identifica o documento que continha um erro:

```
{
    "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."
      }
    ]
}
```

Para obter mais informações sobre as operações síncronas de API, consulte Lote em tempo real APIs.

Processamento em lote assíncrono

Para analisar documentos grandes e coleções grandes de documentos, use as operações assíncronas do Amazon Comprehend.

Para analisar uma coleção de documentos, normalmente você seguirá as seguintes etapas:

1. Armazene os documentos em um bucket do Amazon S3.
2. Inicie um ou mais trabalhos de análise para analisar os documentos.
3. Monitore o progresso dos trabalhos de análise.
4. Recupere os resultados da análise de um bucket do S3 após a conclusão do trabalho.

Para obter mais informações sobre como usar as operações assíncronas de API, consulte Executar tarefas de análise usando o console (console) e Trabalhos de análise assíncrona usando a API.

Idiomas compatíveis no Amazon Comprehend

O Amazon Comprehend é compatível com uma ampla variedade de idiomas para seus diversos atributos. Os idiomas suportados e os atributos que os suportam estão disponíveis nas tabelas a seguir.

Tópicos

- Idiomas compatíveis
- Idiomas compatíveis com atributos do Amazon Comprehend

Idiomas compatíveis

O Amazon Comprehend (exceto o atributo de detecção de idioma dominante) oferece suporte aos seguintes idiomas para um ou mais atributos.

Código	Idioma
de	Alemão
en	Inglês
es	Espanhol
it	Italiano
pt	Português
fr	Francês
ja	Japonês
ko	Coreano
hi	Hindi
ar	Árabe
zh	Chinês (simplificado)

Código	Idioma
zh-TW	Chinês (tradicional)

> **Note**
>
> O Amazon Comprehend identifica o idioma usando identificadores segundo o RFC 5646: se houver um identificador ISO 639-1 de 2 letras, com uma subtag regional./ Se necessário, ele usará essas informações. Caso contrário, ele usará o código ISO 639-2 de 3 letras.
> Para obter mais informações sobre o RFC 5646, consulte Tags para identificação de idiomas no site IETF Tools.

Idiomas compatíveis com atributos do Amazon Comprehend

Atributo	Idiomas compatíveis
Idioma dominante	Consulte Idioma dominante.
Entidades	Todos os idiomas compatíveis.
Frases chave	Todos os idiomas compatíveis.
Detectar entidades de PII	Inglês e espanhol.
Rotular entidades de PII	Inglês e espanhol.
Sentimento	Todos os idiomas compatíveis.
Sentimento direcionado	Inglês.
Análise de sintaxe	Alemão (de), inglês (en), espanhol (es), francês (fr), italiano (it) e português (pt).
Modelagem de tópicos	Não depende do idioma usado. Não oferece suporte a idiomas baseados em caracteres, como chinês, japonês e coreano.

Atributo	Idiomas compatíveis
Classificação personalizada	Os modelos de texto simples oferecem suporte aos seguintes idiomas: alemão (de), inglês (en), espanhol (es), francês (fr), italiano (it) e português (pt). Os modelos de documentos nativos oferecem suporte somente a documentos em inglês.
Reconhecimento de entidade personalizado	Alemão (de), inglês (en), espanhol (es), francês (fr), italiano (it) e português (pt). O reconhecimento personalizado de entidades para PDF e Word é compatível somente com documentos em inglês.

Configuração

Antes de usar o Amazon Comprehend pela primeira vez, execute as seguintes tarefas.

Como configurar tarefas

- Inscreva-se para um Conta da AWS
- Criar um usuário com acesso administrativo
- Configure o AWS Command Line Interface (AWS CLI)
- Conceder acesso programático

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Configure o AWS Command Line Interface (AWS CLI)

Você não precisa AWS CLI executar as etapas dos exercícios de introdução. No entanto, alguns dos outros exercícios neste guia a requerem. Se preferir, você pode pular esta etapa e acessar Introdução ao Amazon Comprehend e configurar a AWS CLI mais tarde.

Para instalar e configurar o AWS CLI

1. Instale AWS CLI o. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Instalando ou atualizando a versão mais recente do AWS Command Line Interface

2. Configure o AWS CLI. Para obter instruções, consulte os seguintes tópicos no Guia do usuário do AWS Command Line Interface :

   Configurar a AWS Command Line Interface

Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do AWS Management Console. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de ferramentas AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para o AWS CLI, AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para ferramentas AWS SDKs e ferramentas, consulte Autenticar usando

Qual usuário precisa de acesso programático?	Para	Por
		credenciais de longo prazo no Guia de referência de ferramentas AWS SDKs e ferramentas. • Para isso AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Introdução ao Amazon Comprehend

O exercício a seguir usa o console do Amazon Comprehend para criar e executar uma tarefa assíncrona de detecção de entidades. Este exercício pressupõe que você esteja familiarizado com o Amazon Simple Storage Service (Amazon S3). Para um exemplo mais simples, consulte Análise em tempo real usando os modelos integrados.

Criar uma tarefa de detecção de entidades

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Tarefas de análise e, em seguida, Criar tarefa.
3. Em Configurações da tarefa, dê um nome a ela. O nome deve ser exclusivo em uma região e conta.
4. Em Tipo de análise, escolha Entidades.
5. Em Idioma, escolha o idioma dos documentos de entrada.
6. Em Dados de entrada, em Fonte de dados, escolha Documentos de exemplo. O console define a Localização do S3 como a pasta contendo as amostras públicas.
7. Em Dados de saída, em Localização do S3, cole a URL ou a localização da pasta no Amazon S3 para os arquivos de saída.
8. Na seção Permissões de acesso, selecione Criar um perfil do IAM. O console cria um novo perfil do IAM com as permissões adequadas para que o Amazon Comprehend acesse os buckets de entrada e saída.
9. Quando terminar de preencher o formulário, escolha Criar tarefa para criar e iniciar a tarefa de detecção de tópicos.

   A nova tarefa aparece na lista de tarefas mostrando seu status no campo com essa finalidade. O campo pode ser `IN_PROGRESS` para uma tarefa em processamento, `COMPLETED` para uma que foi concluída com êxito e `FAILED` para uma tarefa com erro.
10. Escolha a tarefa para abrir o Painel detalhes da tarefa.
11. Em Saída, em Localização dos dados de saída, escolha o link para abrir o console do Amazon S3.
12. No console do Amazon S3, escolha Baixar e salve o arquivo de `output.tar.gz`.
13. Descompacte e salve o arquivo como JSON.

14. Consulte the section called “Entidades” para obter uma descrição dos tipos de entidade e dos campos de cada entidade detectada.

Análise usando o console Amazon Comprehend

Use o console do Amazon Comprehend para analisar documentos em tempo real ou para executar tarefas de análise assíncrona.

Usando a análise em tempo real com modelos integrados, é possível reconhecer entidades, extrair frases-chave, detectar o idioma principal, detectar PII, determinar o sentimento, analisar o sentimento direcionado e analisar a sintaxe.

Você pode executar tarefas de análise usando os modelos integrados para encontrar informações como entidades, eventos, frases, idioma principal, sentimento, sentimento direcionado e informações de identificação pessoal (PII). Também pode executar tarefas de modelagem de tópicos.

O console também é compatível com a análise assíncrona e em tempo real usando modelos personalizados. Para ter mais informações, consulte Classificação personalizada e Reconhecimento de entidade personalizado.

Tópicos

- Análise em tempo real usando os modelos integrados
- Executar tarefas de análise usando o console

Análise em tempo real usando os modelos integrados

Use o console do Amazon Comprehend para executar análises em tempo real de um documento de texto codificado em UTF-8. O documento pode ser em inglês ou em um dos outros idiomas compatíveis com o Amazon Comprehend. Os resultados são exibidos no console para que você possa revisar a análise.

Para começar a analisar documentos, faça login AWS Management Console e abra o console do Amazon Comprehend.

Substitua o texto de amostra pelo seu próprio texto e, em seguida, escolher Analisar para obter uma análise do seu texto. Abaixo do texto que está sendo analisado, o painel Resultados mostra mais informações sobre ele.

Execute análises em tempo real usando o modelo integrado

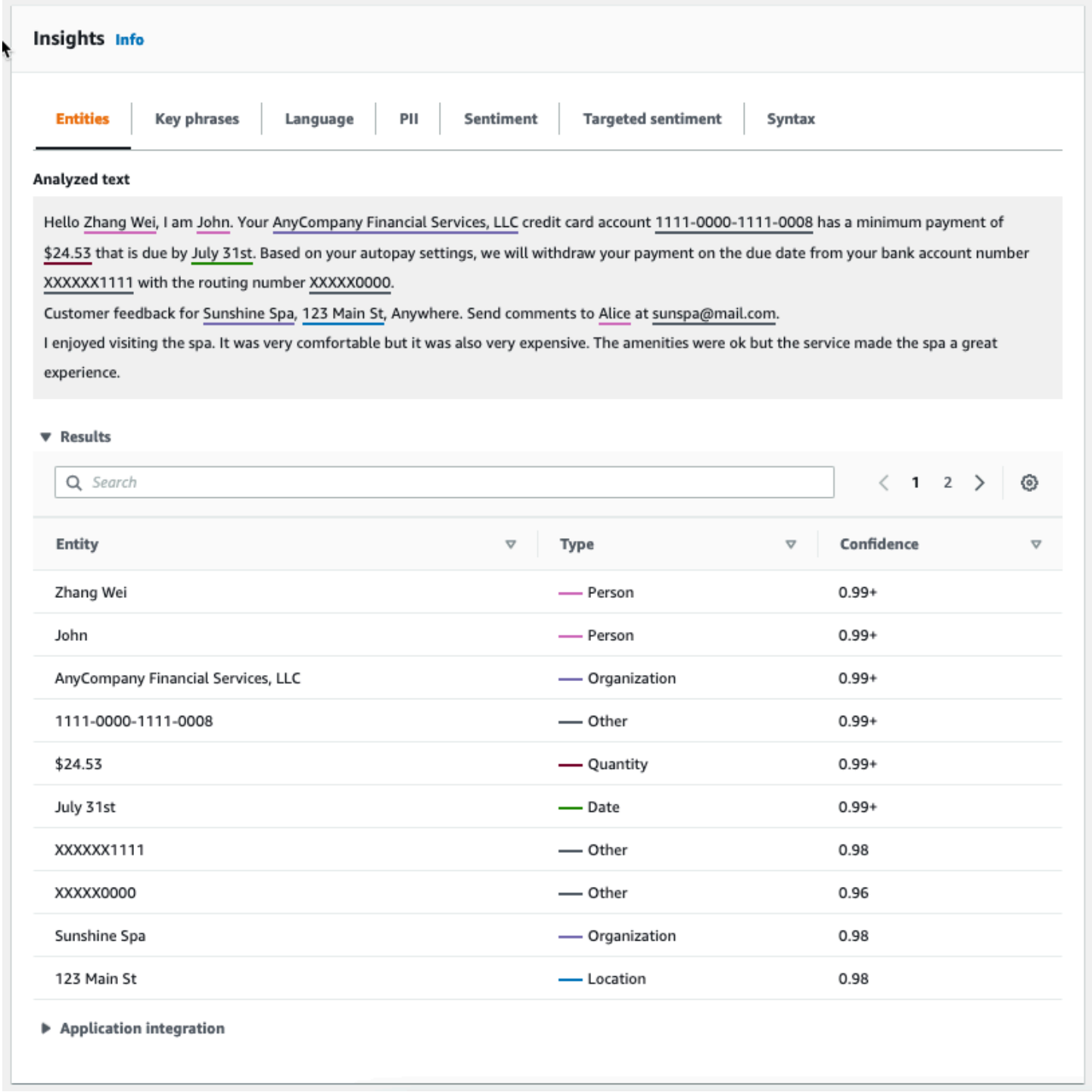
1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Análise em tempo real.
3. Em Tipo de entrada, escolha Incorporar para o Tipo de análise.
4. Insira o texto que você deseja analisar.
5. Escolha Analisar. O console exibe os resultados da análise de texto no painel Insights. O painel Insights inclui uma guia para cada um dos tipos de insights. As seções a seguir descrevem os resultados do tipo de insight.

Tópicos

- Entidades
- Frases chave
- Idioma
- Informações de identificação pessoal (PII)
- Sentimento
- Sentimento direcionado
- Sintaxe

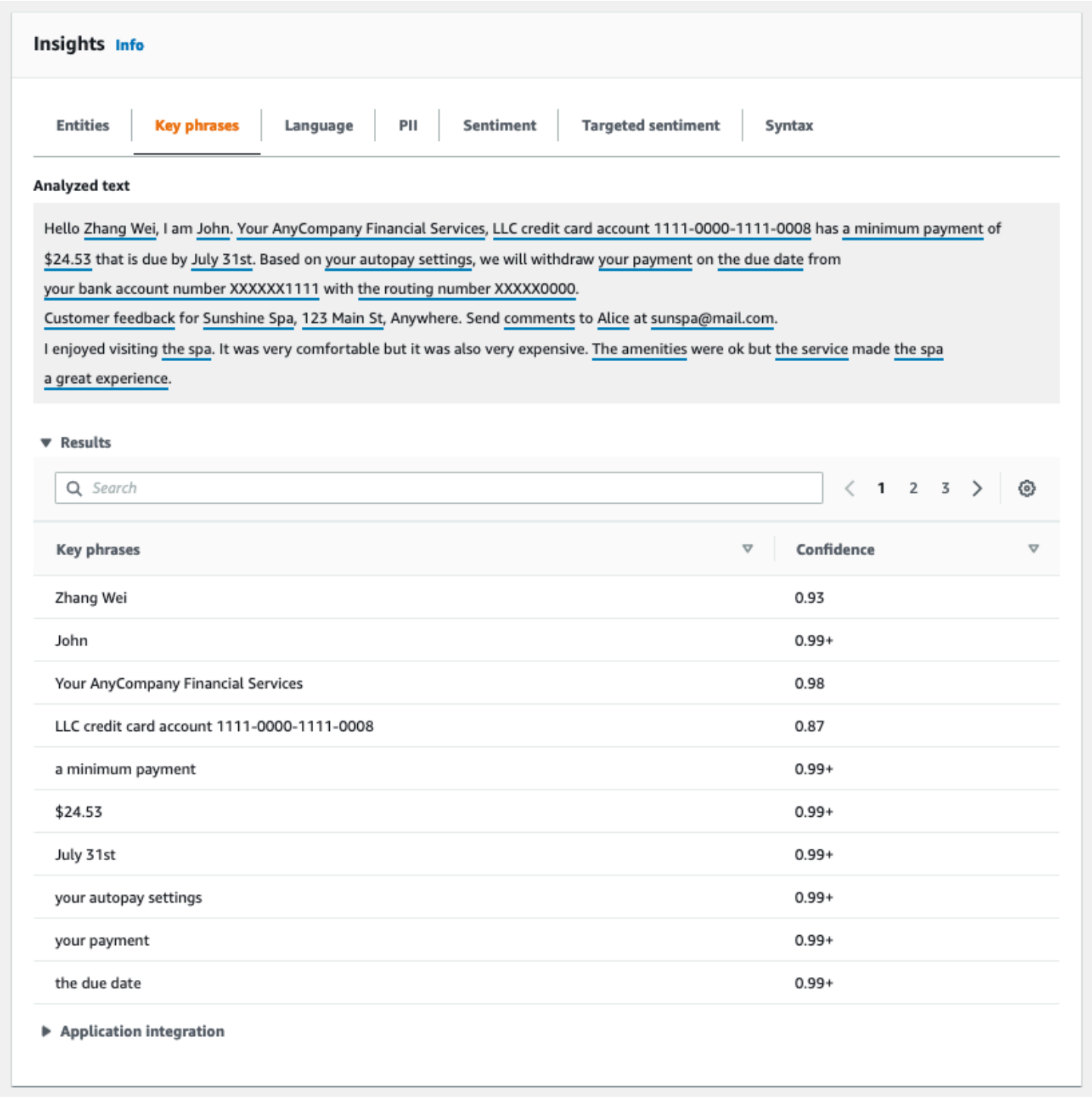
Entidades

A guia Entidades lista cada entidade, sua categoria e o nível de confiança que o Amazon Comprehend detectou no texto de entrada. Os resultados são codificados por cores para indicar diferentes tipos de entidades, como organizações, locais, datas e pessoas. Para obter mais informações, consulte Entidades.



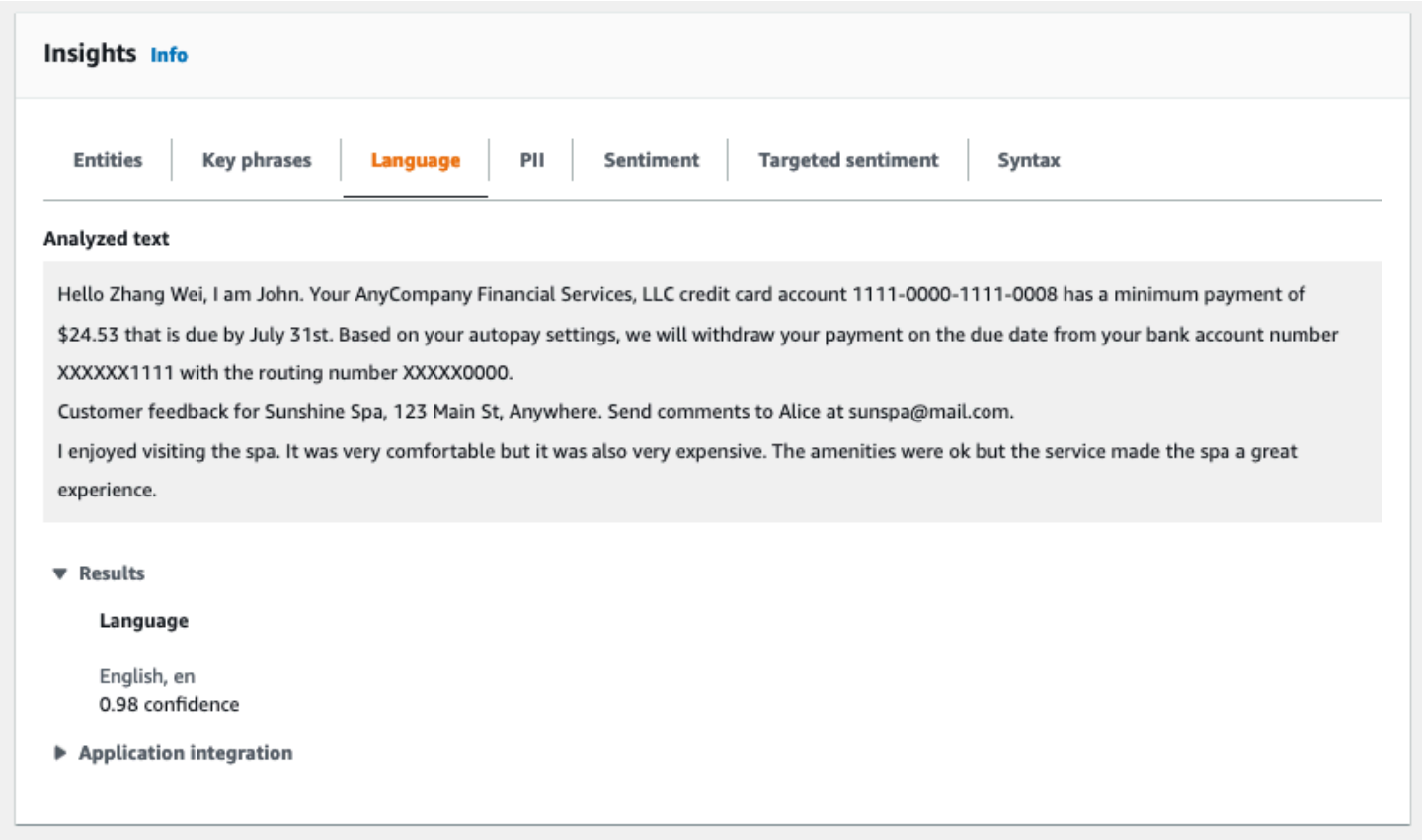
Frases chave

A guia Frases-chave lista as principais frases nominais que o Amazon Comprehend detectou no texto de entrada e o nível de confiança associado. Para obter mais informações, consulte Frases chave.



Idioma

A guia Idioma mostra o idioma dominante do texto e o nível de confiança do Amazon Comprehend de que ele detectou o idioma dominante corretamente. O Amazon Comprehend pode reconhecer 100 idiomas. Para obter mais informações, consulte Idioma dominante.



Informações de identificação pessoal (PII)

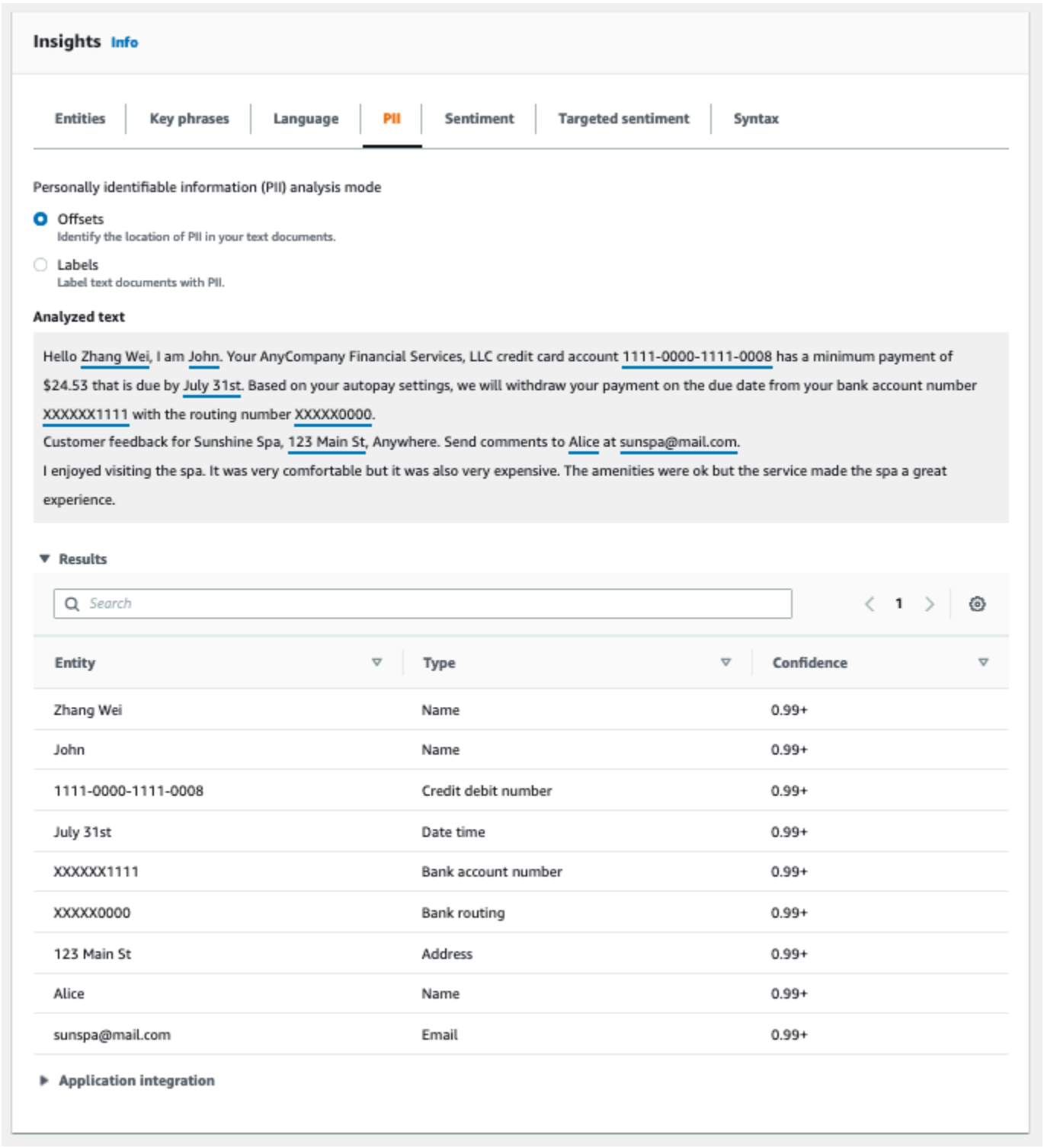
A guia PII lista entidades em seu texto de entrada contendo informações de identificação pessoal (PII). Uma entidade de PII é uma referência textual a dados pessoais que podem ser usados para identificar um indivíduo, como endereço, número de conta bancária ou número de telefone. Para obter mais informações, consulte Detectar entidades de PII.

A guia PII fornece dois modos de análise:

- Offsets
- Rótulos

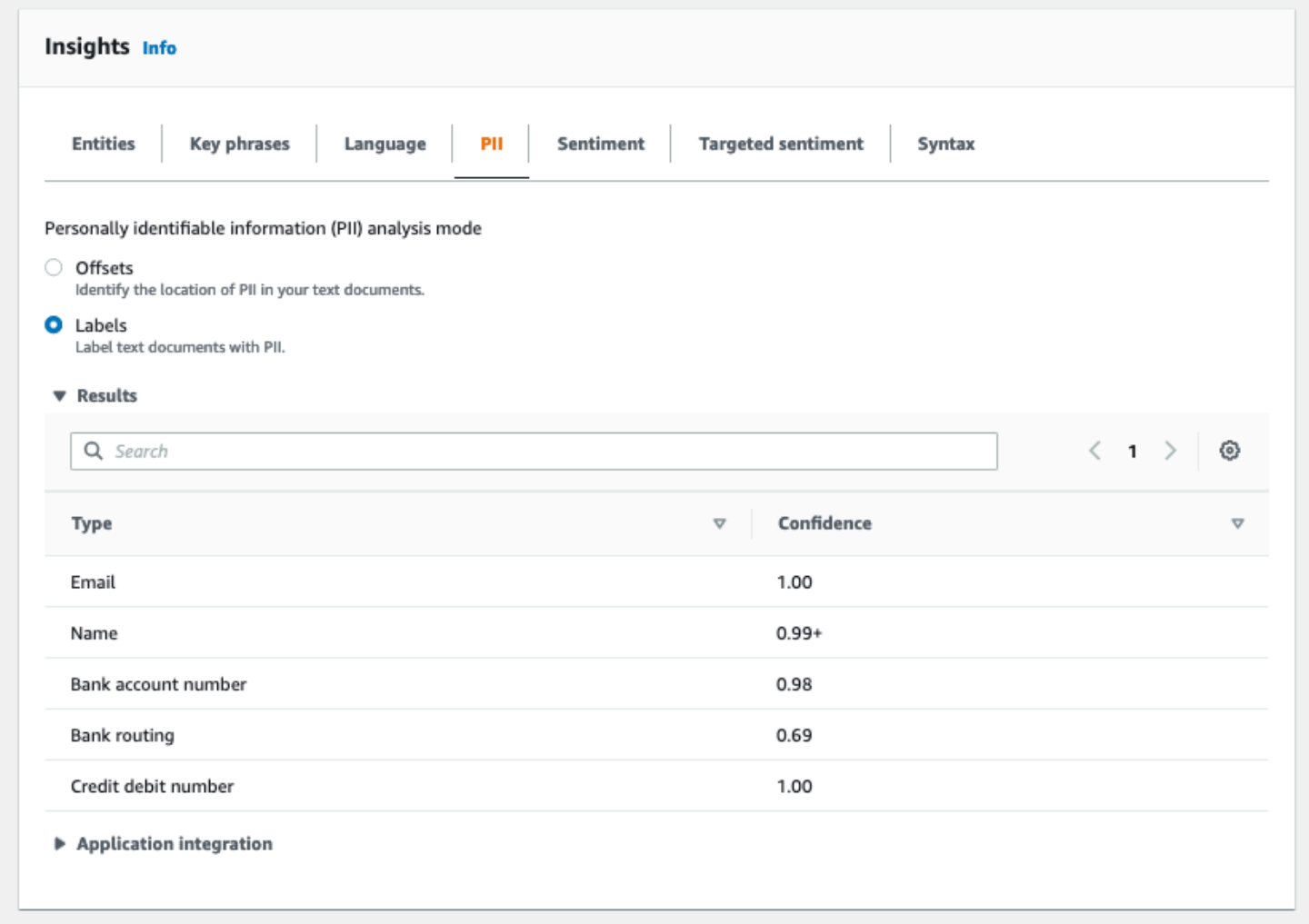
Offsets

O modo de análise de Offsets identifica a localização das PII em seus documentos de texto. Para obter mais informações, consulte Localizar entidades de 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

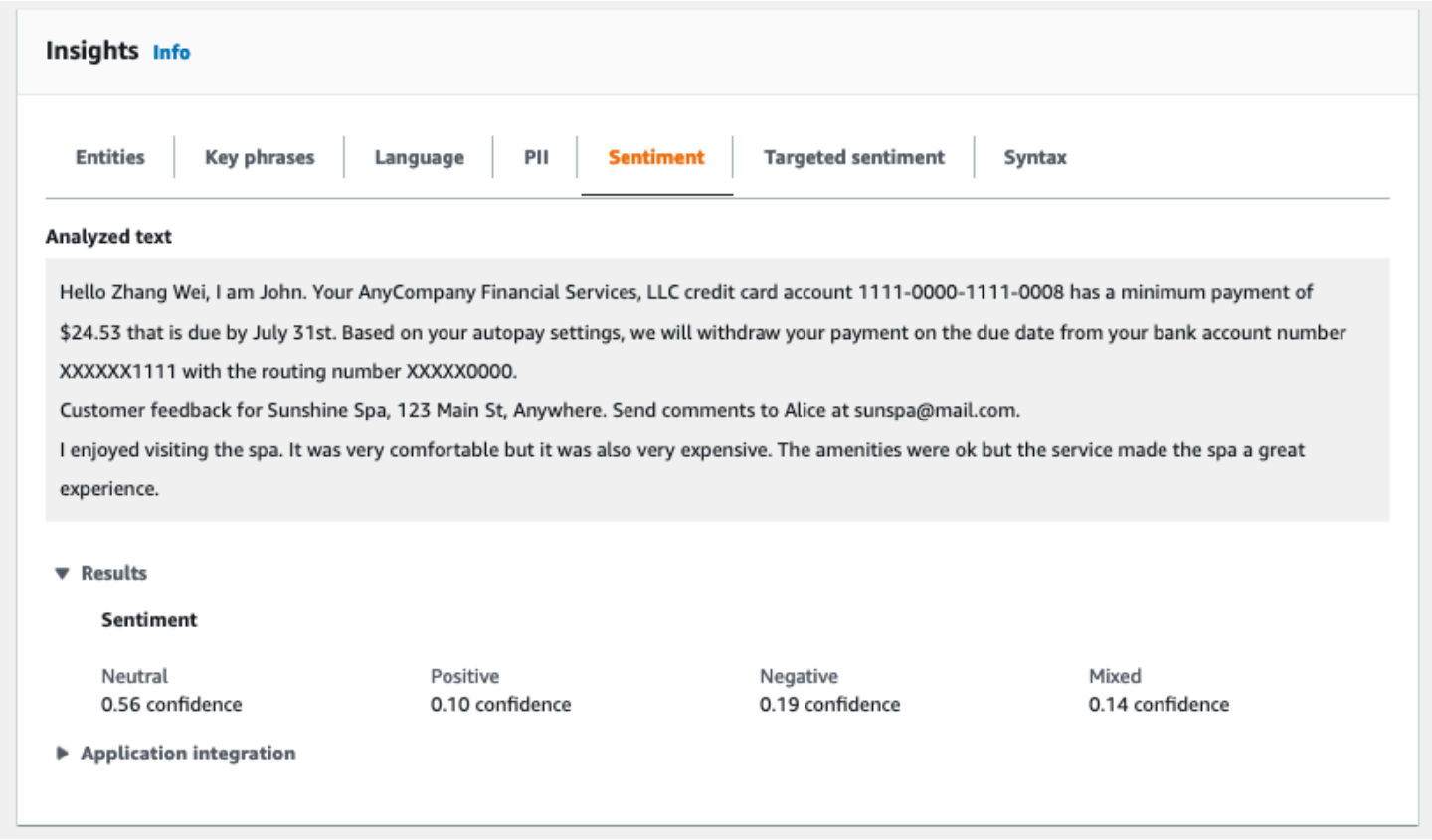
Rótulos

O modo de análise de Rótulos verifica a presença de PII em seu documento de texto e retorna os rótulos dos tipos de entidade de PII identificados. Para obter mais informações, consulte Rotular entidades de PII.



Sentimento

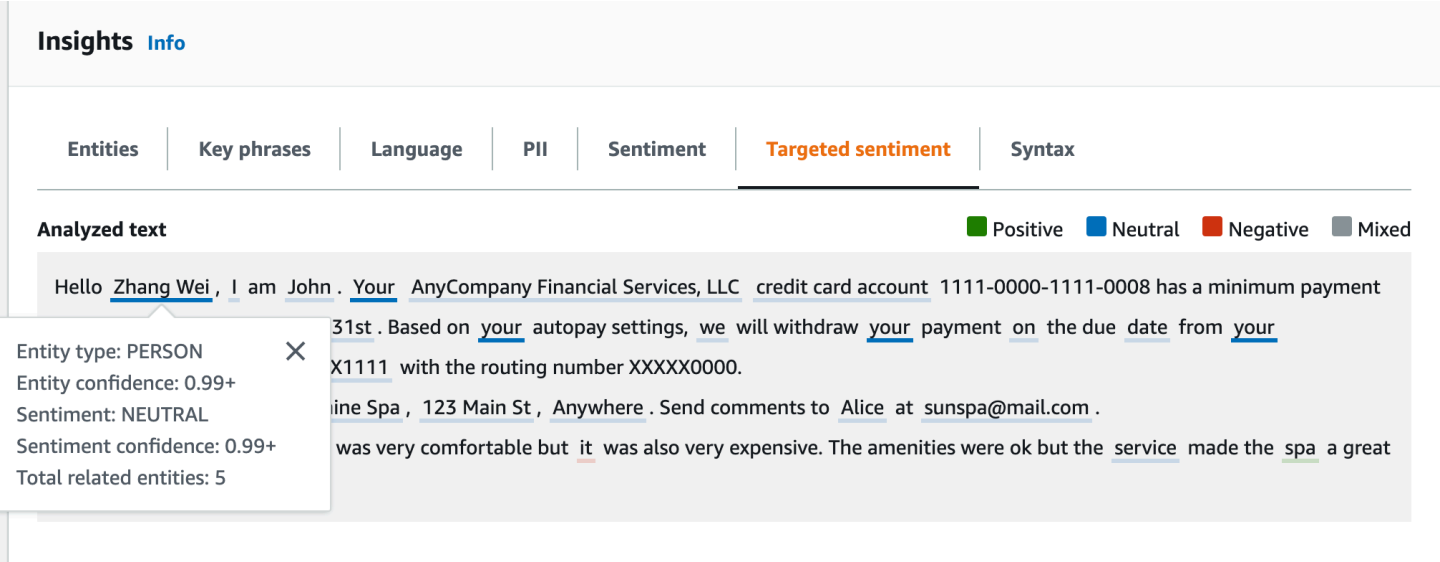
A guia Sentimento mostra o sentimento dominante do texto. O sentimento pode ser classificado como neutro, positivo, negativo ou misto. Nesse caso, cada sentimento tem uma classificação de confiança, fornecendo uma estimativa do Amazon Comprehend de que esse sentimento é dominante. Para obter mais informações, consulte Sentimento.



Sentimento direcionado

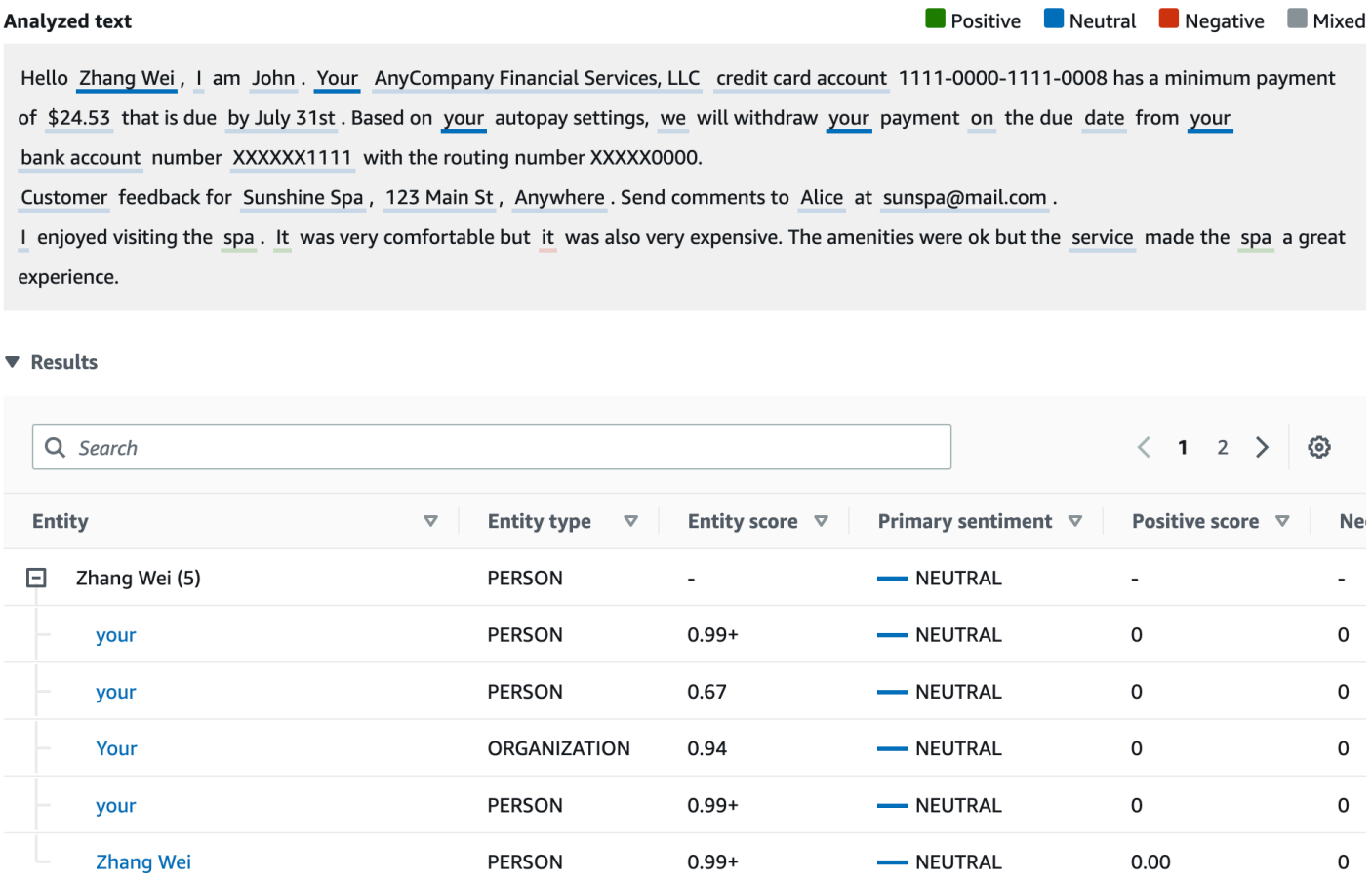
A análise de Sentimento direcionado identifica os sentimentos expressos sobre as entidades mencionadas no texto. O Amazon Comprehend atribui uma classificação de sentimento a cada menção de uma entidade, junto com uma classificação de confiança e outras informações. Uma classificação de sentimento pode ser neutro, positivo, negativo ou misto.

No painel Texto analisado, o console sublinha cada uma das entidades analisadas. A cor do texto sublinhado indica o sentimento geral da entidade. Se você passar o cursor sobre uma entidade, o console exibirá informações adicionais em uma janela pop-up.



A tabela Resultados fornece detalhes adicionais sobre cada entidade. Se houver várias menções da mesma entidade, chamadas de grupo de co-referência, a tabela exibirá essas menções como um conjunto flexível de linhas associadas à entidade principal.

No exemplo a seguir, a entidade é uma pessoa chamada Zhang Wei. A análise de sentimento direcionado reconhece que cada menção sua é uma referência à mesma pessoa. O console exibe essas menções como subentradas da entidade principal.

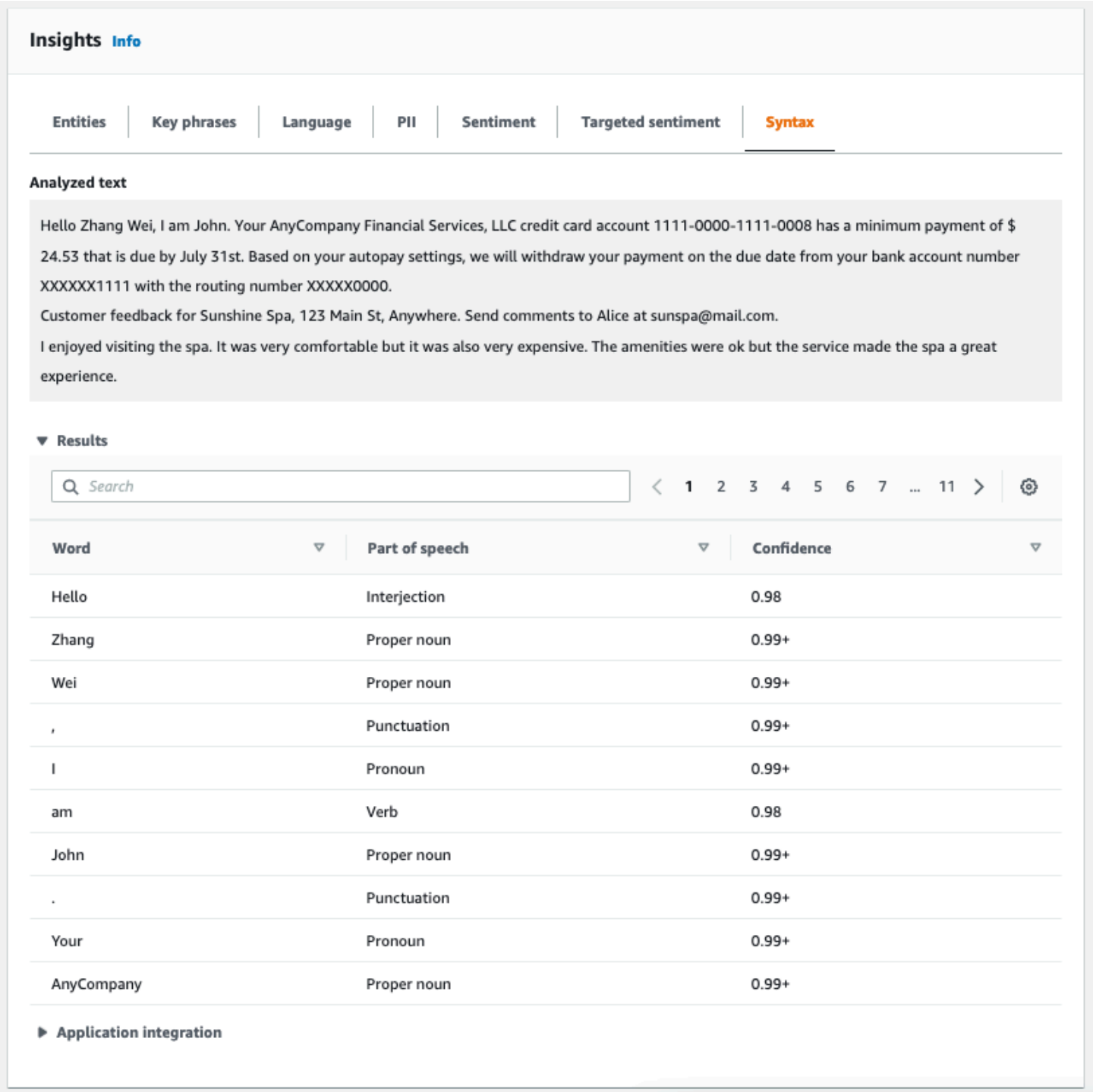


Se o texto que está analisando não incluir nenhum sentimento direcionado Tipos de entidade, a análise exibirá um campo de resultados vazio.

Para mais informações sobre como usar o console para análise do sentimento direcionado em tempo real, consulte Análise em tempo real usando o console.

Sintaxe

A guia Sintaxe mostra um detalhamento de cada elemento no texto, junto com sua parte do discurso e o score de confiança associada. Para obter mais informações, consulte Análise de sintaxe.



Executar tarefas de análise usando o console

Use o console do Amazon Comprehend para criar e gerenciar tarefas de análise assíncronas. Sua tarefa analisa documentos armazenados no Amazon S3 para encontrar entidades como eventos, frases, idioma principal, sentimento ou informações de identificação pessoal (PII).

Criar uma tarefa de análise

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Tarefas de análise e, em seguida, Criar tarefa.
3. Em Configurações da tarefa, dê um nome exclusivo à tarefa de análise.
4. Em Tipo de análise, escolha um dos tipos de análise integrada.

   Se você escolher Linguagem primária ou Modelagem de tópicos, poderá pular a próxima etapa.
5. Dependendo do Tipo de análise escolhido, o console exibe um ou mais dos seguintes campos adicionais:

   - O Idioma é necessário para todos os tipos de análise integrada, exceto o Idioma primário e a Modelagem de tópicos.

     Escolha o idioma dos seus documentos de entrada.
   - Os Tipos de evento alvo são necessários para o tipo de análise de Eventos.

     Selecione os tipos de eventos a serem detectados em seus documentos de entrada. Para informações sobre os tipos de eventos compatíveis, consulte Tipos de eventos.
   - As Configurações de detecção de PII são necessárias para o tipo de análise de PII.

     Selecione o modo de saída. Para mais informações sobre configurações de detecção de PII, consulte Detectar entidades de PII.
6. Em Dados de entrada, especifique onde os documentos de entrada estão localizados no Amazon S3:

   - Para analisar seus próprios documentos, escolha Meus documentos e Procurar no S3 para fornecer o caminho para o bucket ou pasta contendo seus arquivos.
   - Para analisar amostras fornecidas pelo Amazon Comprehend, escolha Documentos de exemplo. Nesse caso, o Amazon Comprehend usa um bucket gerenciado AWS por, e você não especifica a localização.
7. (Opcional) Em Formato de entrada, especifique um dos seguintes formatos para seus arquivos de entrada:

   - Um documento por arquivo: cada arquivo contém um documento de entrada. Essa opção é melhor para coleções de documentos grandes.

- Um documento por linha: a entrada é um ou mais arquivos. Cada linha em um arquivo é considerada um documento. Essa opção é melhor para documentos curtos, como publicações em mídias sociais. Cada linha deve terminar com uma alimentação de linha (LF, \n), um retorno de carro (CR, \r) ou ambos (CRLF, \r\n). Você não pode usar o separador de linha UTF-8 (u+2028) para finalizar uma linha.

8. Em Dados de saída, escolha Procurar no S3. Escolha o bucket ou pasta do Amazon S3 em que você deseja que o Amazon Comprehend grave os dados de saída produzidos pela análise.
9. (Opcional) Para criptografar o resultado de saída da sua tarefa, escolha Criptografia. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta:
   - Se você estiver usando uma chave associada à conta atual, escolha o alias ou ID da chave para o ID da chave KMS.
   - Se você estiver usando uma chave associada a uma conta diferente, insira o ARN do alias ou ID da chave em ID da chave KMS.

   > **Note**
   >
   > Para obter mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte Serviço de gerenciamento de chaves (KMS).

10. Em Permissões de acesso, forneça um perfil do IAM que:
    - Conceda acesso de leitura ao local dos documentos de entrada do Amazon S3.
    - Conceda acesso de gravação ao local dos documentos de saída do Amazon S3.
    - Inclua uma política de confiança que permita a entidade principal do serviço do `comprehend.amazonaws.com` de assumir o perfil e obter suas permissões.

    Se você ainda não tem um perfil do IAM com essas permissões e uma política de confiança apropriada, escolha Criar um perfil do IAM.
11. Quando terminar de preencher o formulário, escolha Criar tarefa para criar e iniciar a tarefa de detecção de tópicos.

A nova tarefa aparece na lista de tarefas mostrando seu status no campo com essa finalidade. O campo pode ser `IN_PROGRESS` para uma tarefa em processamento, `COMPLETED` para uma que foi

concluída com êxito e `FAILED` para uma tarefa com erro. Você pode clicar em uma tarefa para obter mais informações sobre ela, incluindo mensagens de erro.

Quando a tarefa é concluída, o Amazon Comprehend armazena os resultados da análise no local de saída do Amazon S3 especificado para a tarefa. Para obter uma descrição dos resultados da análise para cada tipo de insight, consulte Insights.

Utilizar a API do Amazon Comprehend

A API do Amazon Comprehend oferece suporte às operações para realizar análises em tempo real (síncronas) e operações para iniciar e gerenciar trabalhos de análise assíncrona.

Você pode usar os operadores da API Amazon Comprehend diretamente, ou você pode usar a CLI ou um dos. SDKs Os exemplos deste capítulo usam a CLI, o SDK Python e o SDK Java.

Para executar os exemplos AWS CLI e Python, você deve instalar o. AWS CLI Para obter mais informações, consulte Configure o AWS Command Line Interface (AWS CLI).

Para executar os exemplos de Java, você deve instalar o AWS SDK para Java. Para obter instruções sobre como instalar o SDK para Java, consulte Configurar AWS o SDK para Java.

Tópicos

- Usando o Amazon Comprehend com um SDK AWS
- Análise em tempo real usando a API
- Trabalhos de análise assíncrona usando a API

Usando o Amazon Comprehend com um SDK AWS

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código

Documentação do SDK	Exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Análise em tempo real usando a API

Os exemplos a seguir demonstram como usar a API Amazon Comprehend para análise em tempo real, usando AWS CLI o, e AWS SDKs o para .NET, Java e Python. Use os exemplos para saber mais sobre a operação síncrona do Amazon Comprehend e como compilar blocos para seus próprios aplicativos.

Os exemplos do .NET nesta seção usam o AWS SDK para .NET. Você pode usar o AWS Toolkit for Visual Studiopara desenvolver AWS aplicativos usando o.NET. Ele inclui modelos úteis e o AWS Explorer para implantar aplicativos e gerenciar serviços. Para uma perspectiva de desenvolvedores do.NET AWS, consulte o AWS guia para desenvolvedores.NET.

Tópicos

- Detectando o idioma dominante
- Detectando entidades nomeadas
- Detectando frases-chave
- Determinando o sentimento
- Análise em tempo real de um sentimento direcionado
- Detectando a sintaxe
- Lote em tempo real APIs

Detectando o idioma dominante

Para determinar o idioma dominante usado no texto, use a DetectDominantLanguageoperação. Para detectar o idioma dominante em até 25 documentos em um lote, use a BatchDetectDominantLanguageoperação. Para obter mais informações, consulte Lote em tempo real APIs.

Tópicos

- Usando o AWS Command Line Interface
- Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Usando o AWS Command Line Interface

O exemplo a seguir demonstra como usar a operação `DetectDominantLanguage` com o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

O Amazon Comprehend responde com o seguinte:

```
{
    "Languages": [
```

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

Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Para exemplos de SDK sobre como determinar o idioma dominante, consulte Use `DetectDominantLanguage` com um AWS SDK ou CLI.

Detectando entidades nomeadas

Para determinar as entidades nomeadas em um documento, use a DetectEntitiesoperação. Para detectar entidades em até 25 documentos em um lote, use a BatchDetectEntitiesoperação. Para obter mais informações, consulte Lote em tempo real APIs.

Tópicos

- Usando o AWS Command Line Interface
- Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Usando o AWS Command Line Interface

O exemplo a seguir demonstra como usar a operação `DetectEntities` com o AWS CLI. É necessário especificar o idioma e o texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

O Amazon Comprehend responde com o seguinte:

```
{
    "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"
}
```

Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Para exemplos de SDK sobre como determinar o idioma dominante, consulte Use `DetectEntities` com um AWS SDK ou CLI.

Detectando frases-chave

Para determinar as frases nominais principais usadas no texto, use a DetectKeyPhrasesoperação. Para detectar as principais frases nominais em até 25 documentos em um lote, use a BatchDetectKeyPhrasesoperação. Para obter mais informações, consulte Lote em tempo real APIs.

Tópicos

- Usando o AWS Command Line Interface
- Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Usando o AWS Command Line Interface

O exemplo a seguir demonstra como usar a operação `DetectKeyPhrases` com o AWS CLI. É necessário especificar o idioma e o texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws comprehend detect-key-phrases \
```

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

O Amazon Comprehend responde com o seguinte:

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

Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Para exemplos de SDKs que detectam frases-chave, consulte Use `DetectKeyPhrases` com um AWS SDK ou CLI.

Determinando o sentimento

O Amazon Comprehend fornece as seguintes operações de API para analisar sentimentos:

- DetectSentiment— Determina o sentimento emocional geral de um documento.
- BatchDetectSentiment— Determine o sentimento geral em até 25 documentos em um lote. Para ter mais informações, consulte Lote em tempo real APIs
- StartSentimentDetectionJob— inicia um trabalho assíncrono de detecção de sentimentos para uma coleção de documentos.
- ListSentimentDetectionJobs— Retorna a lista de trabalhos de detecção de sentimentos que você enviou.

- DescribeSentimentDetectionJob— Obtém as propriedades (incluindo status) associadas ao trabalho de detecção de sentimentos especificado.
- StopSentimentDetectionJob— Interrompe o trabalho de sentimento em andamento especificado.

Tópicos

- Usando o AWS Command Line Interface
- Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Usando o AWS Command Line Interface

O exemplo a seguir demonstra como usar a operação `DetectSentiment` com o AWS CLI. Este exemplo especifica o idioma do texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

O Amazon Comprehend responde com o seguinte:

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

Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Para exemplos de SDKs que determinam o sentimento do texto de entrada, consulte Use `DetectSentiment` com um AWS SDK ou CLI.

Análise em tempo real de um sentimento direcionado

O Amazon Comprehend fornece as seguintes operações de API para análise de sentimentos direcionados em tempo real:

- DetectTargetedSentiment— Analisa o sentimento das entidades mencionadas em um documento.
- BatchDetectTargetedSentiment— Analisa o sentimento direcionado de até 25 documentos em um lote. Para ter mais informações, consulte Lote em tempo real APIs

Se o texto que você está analisando não incluir nenhum sentimento direcionado Tipos de entidade, a API retornará uma matriz de entidades vazia.

Usando o AWS Command Line Interface

O exemplo a seguir demonstra como usar a operação `DetectTargetedSentiment` com o AWS CLI. Este exemplo especifica o idioma do texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

O Amazon Comprehend responde com o seguinte:

```
{
"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
          }
        }
      }
    ]
  }
 ]
}
```

Detectando a sintaxe

Para analisar o texto para extrair as palavras individuais e determinar as partes do discurso para cada palavra, use a DetectSyntaxoperação. Para analisar a sintaxe de até 25 documentos em um lote, use a BatchDetectSyntaxoperação. Para obter mais informações, consulte Lote em tempo real APIs.

Tópicos

- Usando AWS Command Line Interface o.
- Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Usando AWS Command Line Interface o.

O exemplo a seguir demonstra como usar a operação `DetectSyntax` com o AWS CLI. Este exemplo especifica o idioma do texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

O Amazon Comprehend responde com o seguinte:

```
{
    "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
        }
    ]
}
```

Usando o AWS SDK para Java SDK para Python ou SDK para .NET

Para exemplos de SDKs que detectam a sintaxe do texto de entrada, consulte Use `DetectSyntax` com um AWS SDK ou CLI.

Lote em tempo real APIs

Para enviar lotes de até 25 documentos, você pode usar as operações em lote em tempo real do Amazon Comprehend. Chamar uma operação em lote é idêntico a chamar o único documento APIs para cada documento na solicitação. Usar o lote APIs pode resultar em melhor desempenho para seus aplicativos. Para obter mais informações, consulte Processamento síncrono de vários documentos.

Tópicos

- Processamento em lote com o AWS CLI
- Processamento em lote com o AWS SDK para .NET

Processamento em lote com o AWS CLI

Esses exemplos mostram como usar operações de API em lote usando o AWS Command Line Interface. Todas as operações, exceto `BatchDetectDominantLanguage`, usam o seguinte arquivo JSON chamado `process.json` como entrada. Para essa operação, a entidade `LanguageCode` não está incluída.

O terceiro documento no arquivo JSON ("$$$$$$$$") causará um erro durante o processamento em lote. Ele é incluído para que as operações incluam um BatchItemErrorna resposta.

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

O exemplo está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Tópicos

- Detecte o idioma dominante usando um batch (AWS CLI)
- Detecte entidades usando um batch (AWS CLI)
- Detecte frases-chave usando um batch (AWS CLI)
- Detecte sentimentos usando um batch (AWS CLI)

Detecte o idioma dominante usando um batch (AWS CLI)

A BatchDetectDominantLanguageoperação determina o idioma dominante de cada documento em um lote. Para obter uma lista dos idiomas que o Amazon Comprehend pode detectar, consulte Idioma dominante. O AWS CLI comando a seguir chama a `BatchDetectDominantLanguage` operação.

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

A seguir está a resposta da operação `BatchDetectDominantLanguage`:

```
{
    "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."
      }
    ]
}
```

Detecte entidades usando um batch (AWS CLI)

Use a BatchDetectEntitiesoperação para encontrar as entidades presentes em um lote de documentos. Para obter mais informações sobre entidades, consulte Entidades. O seguinte comando AWS CLI chama a operação `BatchDetectEntities`.

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

Detecte frases-chave usando um batch (AWS CLI)

A BatchDetectKeyPhrasesoperação retorna as frases nominais principais em um lote de documentos. O AWS CLI comando a seguir chama a `BatchDetectKeyNounPhrases` operação.

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

Detecte sentimentos usando um batch (AWS CLI)

Detecte o sentimento geral de um lote de documentos usando a BatchDetectSentimentoperação. O AWS CLI comando a seguir chama a `BatchDetectSentiment` operação.

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

Processamento em lote com o AWS SDK para .NET

O exemplo de programa a seguir mostra como usar a BatchDetectEntitiesoperação com SDK para .NET o. A resposta do servidor contém um BatchDetectEntitiesItemResultobjeto para cada documento que foi processado com sucesso. Se houver um erro no processamento de um documento, haverá um registro na lista de erros na resposta. O exemplo pega cada um dos documentos com um erro e os reenvia.

O exemplo do .NET nesta seção usam o AWS SDK para .NET. Você pode usar o AWS Toolkit for Visual Studiopara desenvolver AWS aplicativos usando o.NET. Ele inclui modelos úteis e o AWS Explorer para implantar aplicativos e gerenciar serviços. Para uma perspectiva de desenvolvedores do.NET AWS, consulte o AWS guia para desenvolvedores.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");
        }
    }
}
```

Trabalhos de análise assíncrona usando a API

Os exemplos a seguir usam o Amazon Comprehend APIs Asynchronous para criar e gerenciar trabalhos de análise, usando o. AWS CLI

Tópicos

- Análise assíncrona para insights do Amazon Comprehend
- Análise assíncrona para sentimento direcionado

- Análise assíncrona para detecção de eventos
- Análise assíncrona para modelagem de tópicos

Análise assíncrona para insights do Amazon Comprehend

As seções a seguir usam a API do Amazon Comprehend para executar operações assíncronas que analisam os insights do Amazon Comprehend.

Tópicos

- Pré-requisitos
- Iniciar uma tarefa de análise
- Monitorar tarefas de análise
- Obter os resultados da análise

Pré-requisitos

Os documentos devem arquivos de texto no formato UTF-8. Você pode enviar seus documentos em dois formatos. O formato usado depende do tipo de documento que deseja analisar, conforme descrito na tabela a seguir.

Descrição	Formato
Cada arquivo contém um documento de entrada. Essa opção é melhor para coleções de documentos grandes.	Um documento por arquivo
A entrada é um ou mais arquivos. Cada linha em um arquivo é considerada um documento. Essa opção é melhor para documentos curtos, como publicações em mídias sociais. Cada linha deve terminar com uma alimentaç ão de linha (LF, \n), um retorno de carro (CR, \r) ou ambos (CRLF, \r\n). Você não pode usar o separador de linha UTF-8 (u+2028) para finalizar uma linha.	Um documento por linha

Ao iniciar uma tarefa de análise, especifique a localização do S3 para seus dados de entrada. O URI deve estar na mesma AWS região do endpoint da API que você está chamando. O URI pode apontar para um único arquivo ou pode ser o prefixo de uma coleção de arquivos de dados. Para obter mais informações, consulte o tipo de dados InputDataConfig.

É preciso conceder ao Amazon Comprehend acesso ao bucket do Amazon S3 contendo sua coleção de documentos e arquivos de saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Iniciar uma tarefa de análise

Para enviar uma tarefa de análise, use o console do Amazon Comprehend ou a operação de `Start*` apropriada:

- StartDominantLanguageDetectionJob— Comece um trabalho para detectar o idioma dominante em cada documento da coleção. Para mais informações sobre o idioma dominante em um documento, consulte Idioma dominante.
- StartEntitiesDetectionJob— Inicie um trabalho para detectar entidades em cada documento da coleção. Para obter mais informações sobre entidades, consulte Entidades.
- StartKeyPhrasesDetectionJob— Inicie um trabalho para detectar frases-chave em cada documento da coleção. Para mais informações sobre nomes de chave, consulte Frases chave.
- StartPiiEntitiesDetectionJob— Inicie um trabalho para detectar informações de identificação pessoal (PII) em cada documento da coleção. Para mais informações sobre PII, consulte Detectar entidades de PII.
- StartSentimentDetectionJob— Inicie um trabalho para detectar o sentimento em cada documento da coleção. Para mais informações sobre sentimentos, consulte Sentimento.

Monitorar tarefas de análise

A operação de `Start*` retorna um ID que pode ser usado para monitorar o progresso da tarefa.

Para monitorar o progresso usando a API, use uma das duas operações, dependendo se deseja monitorar o progresso de uma tarefa individual ou de várias tarefas.

Para monitorar o progresso de uma tarefa de análise individual, use as operações de `Describe*`. Você fornece o ID da tarefa retornada pela operação de `Start*`. A resposta da operação de `Describe*` contém o campo de `JobStatus` com o status da tarefa.

Para monitorar o progresso de várias tarefas de análise, use a as operações de `List*`. Operações de `List*` retornam uma lista de tarefas que você enviou para o Amazon Comprehend. A resposta inclui um campo de `JobStatus` para cada tarefa que informa o seu status.

Se o campo de status estiver definido como `COMPLETED` ou `FAILED`, o processamento da tarefa foi concluído.

Para obter o status de tarefas individuais, use a operação `Describe*` para a análise que está executando.

- DescribeDominantLanguageDetectionJob
- DescribeEntitiesDetectionJob
- DescribeKeyPhrasesDetectionJob
- DescribePiiEntitiesDetectionJob
- DescribeSentimentDetectionJob

Para obter o status de tarefas múltiplas, use a operação `List*` para a análise que está executando.

- ListDominantLanguageDetectionJobs
- ListEntitiesDetectionJobs
- ListKeyPhrasesDetectionJobs
- ListPiiEntitiesDetectionJobs
- ListSentimentDetectionJobs

Para restringir resultados a tarefas que correspondam a determinados critérios, use o parâmetro `Filter` das operações de `List*`. É possível filtrar os resultados por nome, status da tarefa ou data e hora em que ela foi enviada. Para obter mais informações, consulte o parâmetro `Filter` para cada uma das operações de `List*` na referência de API do Amazon Comprehend.

Obter os resultados da análise

Depois que uma tarefa de análise for concluída, use uma operação `Describe*` para obter a localização dos resultados. Se o status da tarefa for `COMPLETED`, a resposta incluirá um campo `OutputDataConfig` contendo um outro campo com a localização do arquivo de saída no Amazon S3. O arquivo de saída, `output.tar.gz`, é um arquivo compactado contendo os resultados da análise.

Se o status de uma tarefa for `FAILED`, a resposta conterá um campo de `Message` descrevendo o motivo pelo qual a tarefa de análise não foi concluída com êxito.

Para obter o status de tarefas individuais, use a operação `Describe*` apropriada:

- DescribeDominantLanguageDetectionJob
- DescribeEntitiesDetectionJob
- DescribeKeyPhrasesDetectionJob
- DescribeSentimentDetectionJob

Os resultados são retornados em um único arquivo, com uma estrutura JSON para cada documento. Cada arquivo de resposta também inclui mensagens de erro para qualquer tarefa com o campo de status definido como `FAILED`.

Cada uma das seções a seguir mostra exemplos de saída para os dois formatos de entrada.

Obter resultados de detecção de linguagem dominante

Veja a seguir um exemplo de um arquivo de saída de uma análise que detectou o idioma dominante. O formato da entrada é um documento por linha. Para obter mais informações, consulte a operação DetectDominantLanguage.

```
{"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}
```

Veja a seguir um exemplo de saída de uma análise em que o formato da entrada é um documento por arquivo:

```
{"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}]}
```

Obter resultados de detecção de entidades

Veja a seguir um exemplo de um arquivo de saída de uma análise que detectou entidades em documentos. O formato da entrada é um documento por linha. Para obter mais informações, consulte a operação DetectEntities. A saída contém duas mensagens de erro, uma para um documento muito longo e outra para um documento que não está no formato 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."}
```

Veja a seguir um exemplo de saída de uma análise em que o formato da entrada é um documento por arquivo. A saída contém duas mensagens de erro, uma para um documento muito longo e outra para um documento que não está no formato 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."}
```

Obter resultados de detecção de frases-chave

Veja a seguir um exemplo de um arquivo de saída de uma análise que detectou frases-chave em um documento. O formato da entrada é um documento por linha. Para obter mais informações, consulte a operação DetectKeyPhrases.

```
{"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}
```

Veja a seguir um exemplo da saída de uma análise em que o formato da entrada é um documento por arquivo.

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

Obter resultados de detecção de informações de identificação pessoal (PII)

Veja a seguir um exemplo de arquivo de saída de uma tarefa de análise que detectou entidades de PII em documentos. O formato da entrada é um documento por linha.

```
{"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}
```

Veja a seguir um exemplo de saída de uma análise em que o formato da entrada é um documento por arquivo.

```
{"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"}
```

Obter resultados de detecção de sentimento

Veja a seguir um exemplo de um arquivo de saída de uma análise que detectou o sentimento expresso em um documento. Inclui uma mensagem de erro porque um documento é muito longo. O formato da entrada é um documento por linha. Para obter mais informações, consulte a operação DetectSentiment.

```
{"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}}
```

Veja a seguir um exemplo da saída de uma análise em que o formato da entrada é um documento por arquivo.

```
{"File": "small_doc", "Sentiment": "NEUTRAL", "SentimentScore": {"Mixed":
 0.0023450672160834074, "Negative": 0.0009663937962614, "Neutral": 0.9795311689376831,
 "Positive": 0.017157377675175667}}
{"File": "huge_doc", "ErrorCode": "DOCUMENT_SIZE_EXCEEDED", "ErrorMessage": "Document
 size is exceeds the limit of 5120 bytes."}
```

Análise assíncrona para sentimento direcionado

Para obter informações sobre a análise de sentimento direcionado em tempo real, consulte the section called “Análise em tempo real de um sentimento direcionado”.

O Amazon Comprehend fornece as seguintes operações de API para iniciar e gerenciar a análise assíncrona de sentimentos direcionados:

- StartTargetedSentimentDetectionJob— inicia um trabalho assíncrono de detecção de sentimentos direcionados para uma coleção de documentos.
- ListTargetedSentimentDetectionJobs— Retorna a lista de trabalhos direcionados de detecção de sentimentos que você enviou.
- DescribeTargetedSentimentDetectionJob— Obtém as propriedades (incluindo status) associadas ao trabalho de detecção de sentimentos direcionado especificado.
- StopTargetedSentimentDetectionJob— Interrompe o trabalho de sentimento direcionado especificado em andamento.

Tópicos

- Antes de começar
- Como analisar o sentimento direcionado usando o AWS CLI

Antes de começar

Antes de começar, garanta que você tenha:

- Buckets de entrada e saída: identifique os buckets do Amazon S3 que você deseja usar para entrada e saída. O buckets devem estar na mesma região que o API que você está chamando.
- Perfil de serviço do IAM: você deve ter um perfil de serviço do IAM com permissão para acessar seus buckets de entrada e saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Como analisar o sentimento direcionado usando o AWS CLI

O exemplo a seguir demonstra como usar a operação `StartTargetedSentimentDetectionJob` com o AWS CLI. Este exemplo especifica o idioma do texto de entrada.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Para o parâmetro `cli-input-json`, você fornece um caminho para um arquivo JSON que contém os dados da solicitação, conforme exibido no exemplo a seguir.

```
{
    "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"
}
```

Se a solicitação para iniciar o trabalho for bem-sucedida, você receberá a seguinte resposta:

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

Análise assíncrona para detecção de eventos

Tópicos

- Antes de começar
- Detecte eventos usando o AWS CLI

- Listar eventos usando o AWS CLI
- Descreva eventos usando o AWS CLI
- Obter resultados de detecção de eventos

Para detectar eventos em um conjunto de documentos, use o StartEventsDetectionJobpara iniciar um trabalho assíncrono.

Antes de começar

Antes de começar, garanta que você tenha:

- Buckets de entrada e saída: identifique os buckets do Amazon S3 que você deseja usar para entrada e saída. O buckets devem estar na mesma região que o API que você está chamando.
- Perfil de serviço do IAM: você deve ter um perfil de serviço do IAM com permissão para acessar seus buckets de entrada e saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Detecte eventos usando o AWS CLI

O exemplo a seguir demonstra o uso da StartEventsDetectionJoboperação com o AWS CLI

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Para o parâmetro `cli-input-json`, você fornece um caminho para um arquivo JSON que contém os dados da solicitação, conforme exibido no exemplo a seguir.

```
{
  "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"
    ]
}
```

Se a solicitação para iniciar o trabalho de detecção de eventos for bem-sucedida, você receberá a seguinte resposta:

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

Listar eventos usando o AWS CLI

Use a ListEventsDetectionJobsoperação para ver uma lista dos trabalhos de detecção de eventos que você enviou. A lista inclui informações sobre os locais de entrada e saída que você usou, além do status de cada um dos trabalhos de detecção. O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Em resposta, você verá um JSON semelhante a:

```
{
```

```
 "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"
}
```

Descreva eventos usando o AWS CLI

Você pode usar a DescribeEventsDetectionJoboperação para obter o status de um trabalho existente. O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

```
  --job-id job ID
```

Em resposta, você verá o seguinte JSON:

```
{
 "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"
  ]
 }
}
```

Obter resultados de detecção de eventos

Veja a seguir um exemplo de arquivo de saída de um trabalho de análise que detectou eventos em documentos. O formato da entrada é um documento por linha.

```
{"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}
```


Para obter mais informações sobre a estrutura do arquivo de saída de eventos e os tipos de evento compatíveis, consulte Eventos.

Análise assíncrona para modelagem de tópicos

Para determinar os tópicos em um conjunto de documentos, use o StartTopicsDetectionJobpara iniciar um trabalho assíncrono. Você pode monitorar tópicos em documentos escritos em inglês ou espanhol.

Tópicos

- Antes de começar
- Usando o AWS Command Line Interface
- Usando o SDK para Python ou SDK para .NET

Antes de começar

Antes de começar, garanta que você tenha:

- Buckets de entrada e saída: identifique os buckets do Amazon S3 que você deseja usar para entrada e saída. O buckets devem estar na mesma região que o API que você está chamando.
- Perfil de serviço do IAM: você deve ter um perfil de serviço do IAM com permissão para acessar seus buckets de entrada e saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Usando o AWS Command Line Interface

O exemplo a seguir demonstra o uso da `StartTopicsDetectionJob` operação com o AWS CLI

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
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
```

Para o parâmetro `cli-input-json`, você fornece um caminho para um arquivo JSON que contém os dados da solicitação, conforme exibido no exemplo a seguir.

```
{
    "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"
}
```

Se a solicitação para iniciar o trabalho de detecção de tópicos for bem-sucedida, você receberá a seguinte resposta:

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

Use a ListTopicsDetectionJobsoperação para ver uma lista dos trabalhos de detecção de tópicos que você enviou. A lista inclui informações sobre os locais de entrada e saída que você usou, além do status de cada um dos trabalhos de detecção. O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Em resposta, você verá um JSON semelhante a:

```
{
    "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"
            }
        }
    ]
}
```

Você pode usar a DescribeTopicsDetectionJoboperação para obter o status de um trabalho existente. O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Em resposta, você verá o seguinte JSON:

```
{
    "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
    }
}
```

Usando o SDK para Python ou SDK para .NET

Para exemplos de SDK de como iniciar um trabalho de modelagem de tópicos, consulte Use `StartTopicsDetectionJob` com um AWS SDK ou CLI.

Confiança e segurança

Os usuários geram grandes quantidades de conteúdo de texto por meio de aplicativos on-line (como peer-to-peer bate-papos e discussões em fóruns), comentários publicados em sites e por meio de aplicativos de IA generativa (solicitações de entrada e saídas de modelos generativos de IA). Os recursos de confiança e segurança do Amazon Comprehend podem ajudar você a moderar esse conteúdo, a fim de fornecer um ambiente seguro e inclusivo para seus usuários.

Os benefícios de usar os recursos de confiança e segurança do Amazon Comprehend incluem:

- Moderação mais rápida: modere com rapidez e precisão um grande volume de texto para manter suas plataformas online livres de conteúdo impróprio.
- Personalizável: personalize os limites de moderação nas respostas da API para atender às necessidades do seu aplicativo.
- Fácil de usar: configure os recursos de confiança e segurança por meio da LangChain integração ou usando o AWS CLI ou SDKs.

A confiança e a segurança do Amazon Comprehend abordam os seguintes aspectos da moderação de conteúdo:

- Toxicity detection: detectar conteúdo que possa ser prejudicial, ofensivo ou impróprio. Entre eles estão discursos de ódio, ameaças ou abusos.
- Intent classification: detectar conteúdo com intenção maliciosa explícita ou implícita. Entre eles estão conteúdo discriminatório ou ilegal, ou conteúdo que expressa ou solicita aconselhamento sobre assuntos médicos, legais, políticos, controversos, pessoais ou financeiros.
- Privacy protection: os usuários podem fornecer, inadvertidamente, conteúdo que possa revelar informações de identificação pessoal (PII). O Amazon Comprehend PII fornece a capacidade de detectar e editar PII.

Tópicos

- Detecção de toxicidade
- Classificação de segurança imediata
- Detecção e redação de PIIs

Detecção de toxicidade

A detecção de toxicidade do Amazon Comprehend fornece detecção em tempo real de conteúdo tóxico em interações baseadas em texto. Você pode usar a detecção de toxicidade para moderar peer-to-peer conversas em plataformas on-line ou para monitorar entradas e saídas generativas de IA.

A detecção de toxicidade detecta as seguintes categorias de conteúdo ofensivo:

GRAPHIC

A fala gráfica usa imagens visualmente descritivas, detalhadas e desagradavelmente vívidas. Essa linguagem geralmente é detalhada para amplificar um insulto, desconforto ou dano ao destinatário.

HARASSMENT_OR_ABUSE

A fala que impõe uma dinâmica de poder disruptiva entre o falante e o ouvinte, independentemente da intenção, busca afetar o bem-estar psicológico do destinatário ou objetifica uma pessoa.

HATE_SPEECH

Discurso que critica, insulta, denuncia ou desumaniza uma pessoa ou um grupo com base em uma identidade, seja ela raça, etnia, gênero, religião, orientação sexual, habilidade, origem nacional ou outro grupo de identidade.

INSULT

Discurso que inclui linguagem humilhante, zombeteira, insultante ou depreciativa.

PROFANITY

Discurso que contém palavras, frases ou acrônimos indelicados, vulgares ou ofensivos é considerado profano.

SEXUAL

Discurso que indica interesse, atividade ou excitação sexual usando referências diretas ou indiretas a partes do corpo, características físicas ou sexo.

VIOLENCE_OR_THREAT

Discurso que inclui ameaças que buscam infligir dor, ferimentos ou hostilidade a uma pessoa ou grupo.

TOXICITY

Discurso que contém palavras, frases ou acrônimos que podem ser considerados tóxicos em qualquer uma das categorias acima.

Detecção de conteúdo tóxico usando a API

Para detectar conteúdo tóxico no texto, use a DetectToxicContentoperação síncrona. Essa operação executa a análise em uma lista de cadeias de texto que você fornece como entrada. A resposta da API contém uma lista de resultados que corresponde ao tamanho da lista de entrada.

Atualmente, a detecção de conteúdo tóxico é compatível somente com o idioma inglês. Para texto de entrada, você pode fornecer uma lista de até 10 strings de texto. Cada string tem um tamanho máximo de 1 KB.

A detecção de conteúdo tóxico retorna uma lista dos resultados da análise, uma entrada na lista para cada sequência de entrada. Uma entrada contém uma lista dos tipos de conteúdo tóxico identificados na string de texto, junto com uma pontuação de confiança para cada tipo de conteúdo. A entrada também inclui uma pontuação de toxicidade para a string.

Os exemplos a seguir mostram como usar a operação `DetectToxicContent` usando AWS CLI e Python.

AWS CLI

Você pode detectar conteúdo tóxico usando o seguinte comando no AWS CLI:

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

O AWS CLI responde com o seguinte resultado. O segmento de texto recebe uma alta pontuação de confiança na categoria `INSULT`, resultando em uma alta pontuação de toxicidade:

```
{
   "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
        }
    ]
}
```

Você pode inserir até 10 strings de texto, usando o seguinte formato para o parâmetro `text-segments`:

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

Ele AWS CLI responde com os seguintes resultados:

```
{
```

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

Python (Boto)

O exemplo a seguir demonstra como detectar conteúdo tóxico usando o 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)
```

Classificação de segurança imediata

O Amazon Comprehend fornece um classificador binário pré-treinado para classificar solicitações de entrada de texto simples para grandes modelos de linguagem (LLM) ou outros modelos de IA generativa.

O classificador de segurança imediata analisa a solicitação de entrada e atribui uma pontuação de confiança ao fato de a solicitação ser segura ou insegura.

Uma solicitação não segura é uma solicitação de entrada que expressa intenção maliciosa, como solicitar informações pessoais ou privadas, gerar conteúdo ofensivo ou ilegal ou solicitar aconselhamento sobre assuntos médicos, jurídicos, políticos ou financeiros.

Classificação de segurança imediata usando a API

Para executar uma classificação de segurança imediata para uma sequência de texto, use a ClassifyDocumentoperação síncrona. Para entrada, você fornece uma string de texto simples em inglês. A string tem um tamanho máximo de 10 KB.

A resposta inclui duas classes (SAFE e UNSAFE), junto com uma pontuação de confiança para cada classe. O intervalo de valores da pontuação é de zero a um, sendo um indicador de maior confiança.

Os exemplos a seguir mostram como usar a classificação de segurança imediata com o AWS CLI e o Python.

AWS CLI

O exemplo a seguir demonstra como usar o classificador de segurança imediato com 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.'
```

O AWS CLI responde com a seguinte saída:

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

> **Note**
>
> Ao usar o `classify-document` comando, para o `--endpoint-arn` parâmetro, você deve passar um ARN que use o mesmo que sua Região da AWS AWS CLI configuração. Para configurar o AWS CLI, execute o `aws configure` comando. Neste exemplo, o ARN do endpoint tem o código da região `us-west-2`. Você pode usar o classificador de segurança imediato em qualquer uma das seguintes regiões:
>
> - us-east-1
> - us-west-2
> - eu-west-1
> - ap-southeast-2

Python (Boto)

O exemplo a seguir demonstra como usar o classificador de segurança imediata com 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**
>
> Ao usar o método `classify_document`, para o argumento `EndpointArn`, você deve passar um ARN que use a mesma Região da AWS que seu SDK cliente boto3. Neste exemplo, o ARN do cliente e do endpoint usam `us-west-2`. Você pode usar o classificador de segurança imediato em qualquer uma das seguintes regiões:
>
> - us-east-1
> - us-west-2
> - eu-west-1

- ap-southeast-2

Detecção e redação de PIIs

Você pode usar o console do Amazon Comprehend APIs ou detectar informações de identificação pessoal (PII) em documentos de texto em inglês ou espanhol. PII é uma referência textual a dados pessoais que pode identificar um indivíduo. Exemplos de PII incluem endereços, números de contas bancárias e números de telefone.

Você pode detectar ou redigir as entidades PII no texto. Para detectar entidades de PII, você pode usar a análise em tempo real ou uma tarefa em lotes assíncrono. Para redigir as entidades de PII, você deve usar uma tarefa em lotes assíncrono.

Para obter mais informações, consulte Informações de identificação pessoal (PII) .

Informações de identificação pessoal (PII)

Você pode usar o console do Amazon Comprehend APIs ou detectar informações de identificação pessoal (PII) em documentos de texto em inglês ou espanhol. PII é uma referência textual a dados pessoais que pode se usada para identificar uma pessoa. Exemplos de PII incluem endereços, números de contas bancárias e números de telefone.

Com a detecção de PII, você tem a opção de localizar as entidades de PII ou redigir as entidades de PII no texto. Para localizar entidades de PII, você pode usar a análise em tempo real ou uma tarefa em lotes assíncrono. Para redigir as entidades de PII, você deve usar uma tarefa em lotes assíncrono.

Você pode usar os pontos de acesso do Amazon S3 Object Lambda para informações de identificação pessoal (PII) a fim de controlar a recuperação de documentos do seu bucket do Amazon S3. Você pode controlar o acesso a documentos que contêm PII e redigir informações de identificação pessoal dos documentos. Para obter mais informações, consulte Utilizar pontos de acesso do Amazon S3 object Lambda para informações de identificação pessoal (PII).

Tópicos

- Detectar entidades de PII
- Rotular entidades de PII
- Análise em tempo real de PII (console)
- Tarefas de análise assíncrona de PII (console)
- Análise em tempo real (API) de PII
- Tarefas de análise assíncrona de PII (API)

Detectar entidades de PII

Você pode usar o Amazon Comprehend para detectar entidades de PII em documentos de texto em inglês ou espanhol. Uma entidade PII é um tipo específico de informações de identificação pessoal (PII). Use a detecção de PII para localizar as entidades de PII ou redigir as entidades de PII no texto.

Tópicos

- Localizar entidades de PII
- Redigir entidades de PII

- Tipos de entidades universais de PII
- Tipos de entidade de PII específicos de cada país

Localizar entidades de PII

Para localizar as entidades de PII em seu texto, você pode analisar rapidamente um único documento usando análise em tempo real, além de iniciar uma tarefa em lotes assíncronos em uma coleção de documentos.

Use o console ou a API para análise em tempo real de um único documento. Seu texto de entrada pode incluir até 100 kilobytes de caracteres codificados em UTF-8.

Por exemplo, você pode enviar o seguinte texto de entrada para localizar as entidades de PII:

Olá Paulo Santos. O extrato mais recente da sua conta de cartão de crédito 1111-0000-1111-0000 foi enviado para 123 Any Street, Seattle, WA 98109.

A saída inclui as informações de que “Paul Santos” tem o tipo `NAME`, “1111-0000-1111-0000" tem o tipo `CREDIT_DEBIT_NUMBER`, e “123 Any Street, Seattle, WA 98109" tem o tipo `ADDRESS`.

O Amazon Comprehend retorna uma lista de entidades de PII detectadas, com as seguintes informações para cada entidade de PII:

- Uma pontuação que estima a probabilidade de que a extensão de texto detectada seja do tipo de entidade detectada.
- O tipo de entidade PII.
- A localização da entidade PII no documento, especificada como offsets de caracteres para o início e o fim da entidade.

Por exemplo, o texto de entrada mencionado anteriormente produz a seguinte resposta:

```
{
    "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
        }
    ]
}
```

Redigir entidades de PII

Para redigir as entidades de PII em seu texto, use o console ou a API para iniciar uma tarefa em lote assíncrono. O Amazon Comprehend retorna uma cópia do texto de entrada com redações para cada entidade de PII.

Por exemplo, você pode enviar o seguinte texto de entrada para redigir as entidades de PII:

Olá Paulo Santos. O extrato mais recente da sua conta de cartão de crédito 1111-0000-1111-0000 foi enviado para 123 Any Street, Seattle, WA 98109.

O arquivo de saída inclui o seguinte texto:

Olá ***** ******. O extrato mais recente da sua conta de cartão de crédito ********** foi enviado para *** ********* ********** ** *******.

Tipos de entidades universais de PII

Alguns tipos de entidades de PII são universais (não específicos para países individuais), como endereços de e-mail e números de cartão de crédito. O Amazon Comprehend detecta os seguintes tipos de entidades universais de PII:

ADDRESS

Um endereço físico, como “100 Main Street, Anytown, EUA” ou “Suíte #12, Edifício 123”. Um endereço pode incluir informações como rua, prédio, localização, cidade, estado, país, condado, código postal, distrito e bairro.

AGE

A idade de uma pessoa, incluindo a quantidade e a unidade de tempo. Por exemplo, na frase “Tenho 40 anos”, o Amazon Comprehend reconhece “40 anos” como uma idade.

AWS_ACCESS_CHAVE

Um identificador exclusivo associado a uma chave de acesso secreta; você usa o ID da chave de acesso e a chave de acesso secreta para assinar AWS solicitações programáticas criptograficamente.

AWS_SECRET_CHAVE

Um identificador exclusivo associado a uma chave de acesso. Você usa o ID da chave de acesso e a chave de acesso secreta para assinar AWS solicitações programáticas criptograficamente.

CREDIT_DEBIT_CVV

Um código de verificação de cartão (CVV) de três dígitos que está presente nos cartões de crédito e débito VISA e Discover. MasterCard Para cartões de crédito ou débito American Express, o CVV é um código numérico de quatro dígitos.

CREDIT_DEBIT_EXPIRY

Data de validade do cartão de crédito ou débito. Esse número geralmente tem quatro dígitos e geralmente é formatado como month/year ou MM/AA. O Amazon Comprehend reconhece datas de expiração como 01/21, 01/2021 e jan de 2021.

CREDIT_DEBIT_NUMBER

O número de um cartão de crédito ou débito. Esses números podem variar de 13 a 16 dígitos. No entanto, o Amazon Comprehend também reconhece números de cartão de crédito ou débito quando somente os últimos quatro dígitos estão presentes.

DATE_TIME

Uma data pode incluir um ano, mês, dia, dia da semana ou hora do dia. Por exemplo, o Amazon Comprehend reconhece “19 de janeiro de 2020" ou “11h” como datas. O Amazon Comprehend reconhecerá datas parciais, conjunto de datas e intervalos de datas. Também reconhecerá décadas, como “a década de 1990”.

DRIVER_ID

O número atribuído a uma carteira de motorista, que é um documento oficial que permite que uma pessoa opere um ou mais veículos motorizados em uma via pública. O número da carteira de motorista consiste em caracteres alfanuméricos.

EMAIL

Um endereço de e-mail, como marymajor@email.com.

INTERNATIONAL_BANK_ACCOUNT_NUMBER

Um número de conta bancária internacional tem formatos específicos em cada país. Consulte www.iban.com/structure.

IP_ADDRESS

Um IPv4 endereço, como 198.51.100.0.

LICENSE_PLATE

A placa de um veículo emitida pelo estado ou país em que o veículo está registrado. O formato para veículos de passageiros normalmente tem de cinco a oito dígitos, consistindo em letras maiúsculas e números. O formato varia de acordo com a localização do estado ou país emissor.

MAC_ADDRESS

Um endereço MAC (controle de acesso à mídia) é um identificador exclusivo atribuído a um controlador de interface de rede (NIC).

NAME

O nome de uma pessoa. Esse tipo de entidade não inclui títulos, como Dr., Sr., Sra. ou Senhorita. O Amazon Comprehend não aplica esse tipo de entidade a nomes que fazem parte de organizações ou endereços. Por exemplo, o Amazon Comprehend reconhece a “Organização John Doe” como uma organização e reconhece a “Rua Jane Doe” como um endereço.

PASSWORD

Uma sequência alfanumérica usada como senha, como “*very20special #pass *”.

PHONE

Um número de telefone. Esse tipo de entidade também inclui números de fax e pager.

PIN

Um número de identificação pessoal (PIN) de quatro dígitos com o qual você pode acessar sua conta bancária.

SWIFT_CODE

Um código SWIFT é um formato padrão do Código identificador bancário (BIC) usado para especificar um determinado banco ou agência. Os bancos usam esses códigos para transferências de dinheiro, como transferências eletrônicas internacionais.

Os códigos SWIFT consistem em oito ou 11 caracteres. Os códigos de 11 dígitos se referem a filiais específicas, enquanto os códigos de oito dígitos (ou códigos de 11 dígitos terminados em 'XXX') se referem à sede ou ao escritório principal.

URL

Um endereço da web, como www.example.com.

USERNAME

Um nome de usuário que identifica uma conta, como nome de login, nome de tela, apelido ou identificador.

VEHICLE_IDENTIFICATION_NUMBER

Um Número de identificação de veículo (VIN) identifica um veículo de forma exclusiva. O conteúdo e o formato do VIN são definidos na especificação ISO 3779. Cada país tem códigos e formatos específicos para VINs.

Tipos de entidade de PII específicos de cada país

Alguns tipos de entidades de PII são específicos de cada país, como números de passaporte e outros números de identificação emitidos pelo governo. O Amazon Comprehend detecta os seguintes tipos de entidades de PII específicas de cada país:

CA_HEALTH_NUMBER

O Canadian Health Service Number (Número do serviço de saúde canadense) é um identificador exclusivo de 10 dígitos, necessário para que as pessoas tenham acesso aos benefícios de saúde.

CA_SOCIAL_INSURANCE_NUMBER

O Canadian Social Insurance Number (SIN: Número do seguro social canadense) é um identificador exclusivo de nove dígitos, necessário para que as pessoas acessem programas e benefícios governamentais.

O SIN é formatado como três grupos de três dígitos, como 123-456-789. Um SIN pode ser validado por meio de um processo simples de verificação de dígitos chamado algoritmo de Luhn.

IN_AADHAAR

Um Aadhaar indiano é um número de identificação exclusivo de 12 dígitos emitido pelo governo indiano para os residentes da Índia. O formato Aadhaar tem um espaço ou hífen após o quarto e oitavo dígitos.

IN_NREGA

Um número da Indian National Rural Employment Guarantee Act (NREGA — Lei nacional de garantia de emprego rural da Índia) consiste em duas letras seguidas por 14 números.

IN_PERMANENT_ACCOUNT_NUMBER

Um número de conta permanente indiano é um número alfanumérico exclusivo de 10 dígitos emitido pelo Departamento de Imposto de Renda.

IN_VOTER_NUMBER

Um ID de eleitor indiano consiste em três letras seguidas por sete números.

UK_NATIONAL_HEALTH_SERVICE_NUMBER

Um UK National Health Service Number (Número do Serviço Nacional de Saúde do Reino Unido) é um número de 10 a 17 dígitos, como 485 777 3456. O sistema atual formata o número de 10 dígitos com espaços após o terceiro e o sexto dígitos. O dígito final é uma soma de verificação que detecta erros.

O formato numérico de 17 dígitos tem espaços após o 10º e o 13º dígitos.

UK_NATIONAL_INSURANCE_NUMBER

Um UK National Insurance Number (NINO: Número de seguro nacional do Reino Unido) que fornece às pessoas acesso aos benefícios do Seguro Nacional (previdência social). Também é usado para alguns fins no sistema tributário do Reino Unido.

O número tem nove dígitos e começa com duas letras, seguidas por seis números e uma letra. Um NINO pode ser formatado com um espaço ou um traço após as duas letras e depois do segundo, quarto e sexto dígitos.

UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER

Uma Unique Taxpayer Reference (UTR — Referência única de contribuinte) do Reino Unido é um número de 10 dígitos que identifica um contribuinte ou uma empresa.

BANK_ACCOUNT_NUMBER

Um número de conta bancária dos EUA, que normalmente tem de 10 a 12 dígitos. O Amazon Comprehend também reconhece números de contas bancárias quando somente os últimos quatro dígitos estão presentes.

BANK_ROUTING

Um número de roteamento de conta bancária dos EUA. Normalmente, eles têm nove dígitos, mas o Amazon Comprehend também reconhece números de roteamento quando somente os últimos quatro dígitos estão presentes.

PASSPORT_NUMBER

Um número de passaporte dos EUA. Os números dos passaportes variam de seis a nove caracteres alfanuméricos.

US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER

Um Número de Identificação Fiscal Individual (ITIN) dos EUA é um número de nove dígitos que começa com um “9” e contém um “7” ou “8” como o quarto dígito. Um ITIN pode ser formatado com um espaço ou um traço após o terceiro e o quarto dígitos.

SSN

O Social Security Number (SSN — Número do seguro social) dos EUA é um número de nove dígitos emitido para cidadãos dos EUA, residentes permanentes e residentes temporários que trabalham. O Amazon Comprehend também reconhece números do seguro social quando somente os últimos quatro dígitos estão presentes.

Rotular entidades de PII

Quando você executa a detecção de PII, o Amazon Comprehend retorna os rótulos dos tipos de entidades de PII identificados. Por exemplo, se você enviar o seguinte texto de entrada para o Amazon Comprehend:

Olá Paulo Santos. O extrato mais recente da sua conta de cartão de crédito 1111-0000-1111-0000 foi enviado para 123 Any Street, Seattle, WA 98109.

A saída inclui rótulos que representam os tipos de entidades de PII junto com uma pontuação de confiança da precisão. Nesse caso, o texto do documento “Paul Santos”, “1111-0000-1111-0000" e “123 Any Street, Seattle, WA 98109" geram os rótulos `NAME`, `CREDIT_DEBIT_NUMBER`, e `ADDRESS`,

respectivamente, como tipos de entidade de PII. Para mais informações sobre os tipos de entidades compatíveis, consulte Tipos de entidades universais de PII.

O Amazon Comprehend fornece as seguintes informações para cada rótulo:

- O nome do rótulo do tipo de entidade PII.
- Uma pontuação que estima a probabilidade de o texto detectado ser rotulado como um tipo de entidade PII.

O exemplo de texto de entrada acima resulta na seguinte saída JSON.

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

Análise em tempo real de PII (console)

Você pode usar o console para executar a detecção de PII em tempo real de um documento de texto. O tamanho máximo do texto é de 100 kilobytes de caracteres codificados em UTF-8. O console exibe os resultados para que você possa revisar a análise.

Execute a análise em tempo real de detecção de PII usando o modelo integrado

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Análise em tempo real.

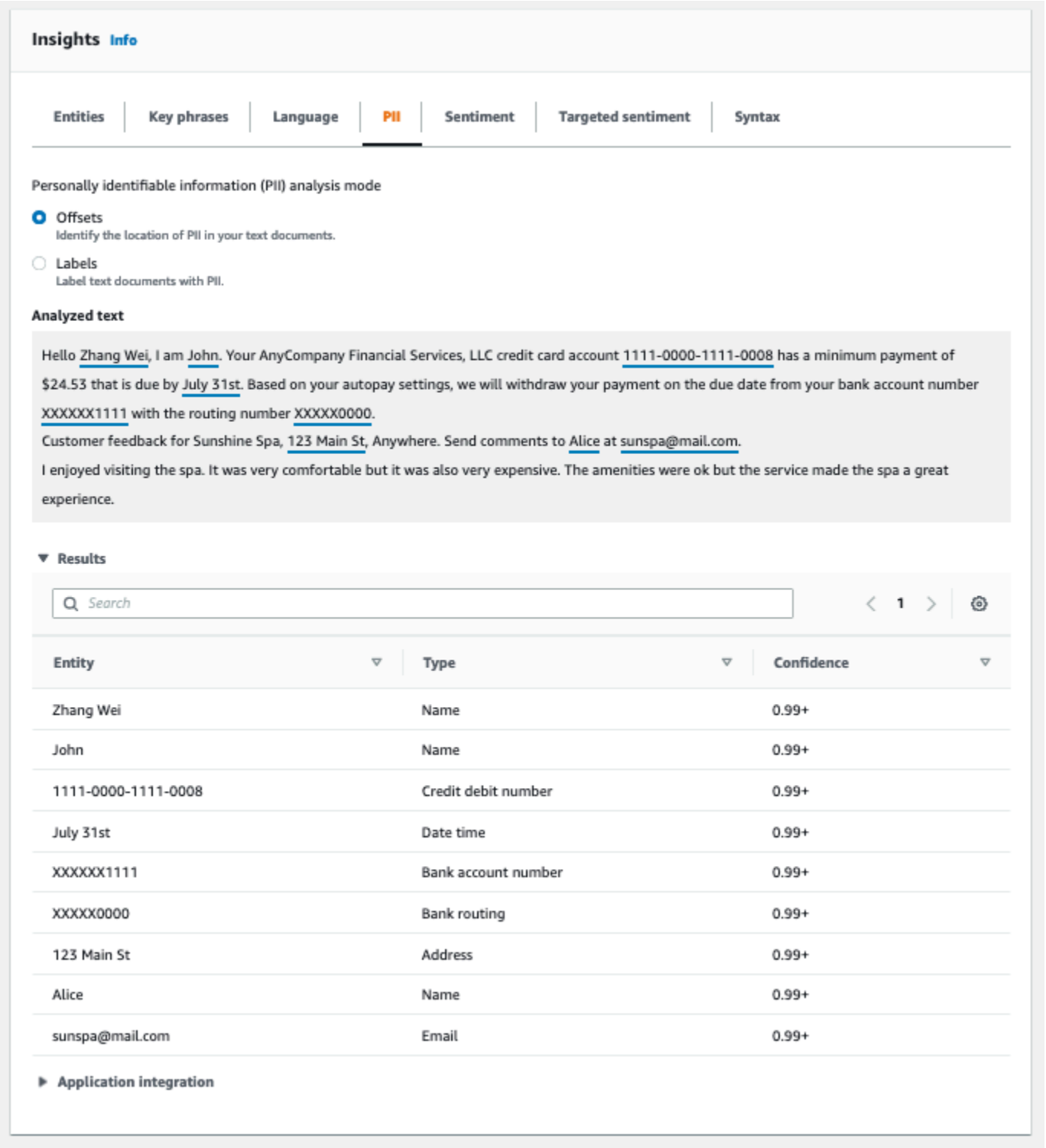
3. Em Tipo de entrada, escolha Incorporar para o Tipo de análise.
4. Insira o texto que você deseja analisar.
5. Escolha Analisar. O console exibe os resultados da análise de texto no painel Insights. A guia PII lista as entidades de PII detectadas no texto de entrada.

No painel Insights, a guia PII exibe os resultados de dois modos de análise:

- Offsets: identifica a localização das PII no documento de texto.
- Rótulos: identifica os rótulos dos tipos de entidades de PII identificados.

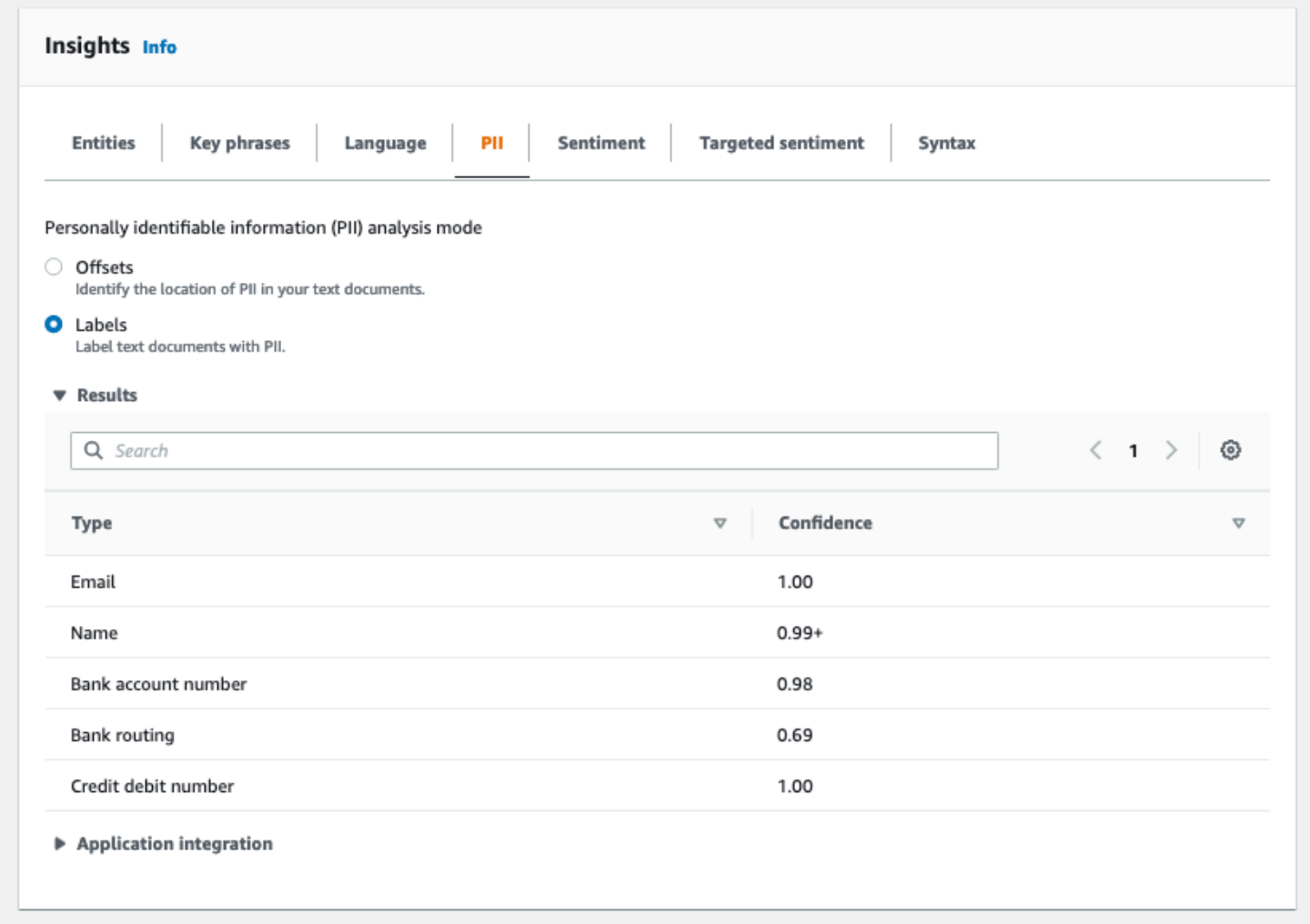
Offsets

O modo de análise de Offsets identifica a localização das PII em seus documentos de texto. Para obter mais informações, consulte Localizar entidades de PII.



Rótulos

O modo de análise de Rótulos retorna os rótulos dos tipos de entidade de PII identificados. Para obter mais informações, consulte Rotular entidades de PII.



Tarefas de análise assíncrona de PII (console)

Você pode usar o console para criar tarefas de análise assíncrona para detectar entidades de PII. Para mais informações sobre os tipos de entidade de PII, consulte Detectar entidades de PII.

Criar uma tarefa de análise

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Tarefas de análise e, em seguida, Criar tarefa.
3. Em Configurações da tarefa, dê um nome exclusivo à tarefa de análise.
4. Para Tipo de análise, escolha Informações de identificação pessoal (PII).
5. Em Idioma, escolha um dos idiomas suportados (inglês ou espanhol).
6. Em Modo de saída, selecione uma das seguintes opções:

- Offsets: a saída da tarefa retorna a localização de cada entidade de PII.
- Redações: a saída da tarefa retorna uma cópia do texto de entrada com cada entrada de PII editada.

7. (Opcional) Se escolher Redações como modo de saída, você pode selecionar os tipos de entidade de PII a serem redigidos.
8. Em Dados de entrada, especifique onde os documentos de entrada estão localizados no Amazon S3:
   - Para analisar seus próprios documentos, escolha Meus documentos e Procurar no S3 para fornecer o caminho para o bucket ou pasta contendo seus arquivos.
   - Para analisar amostras fornecidas pelo Amazon Comprehend, escolha Documentos de exemplo. Nesse caso, o Amazon Comprehend usa um bucket que é gerenciado AWS por, e você não especifica a localização.
9. (Opcional) Em Formato de entrada, especifique um dos seguintes formatos para seus arquivos de entrada:
   - Um documento por arquivo: cada arquivo contém um documento de entrada. Essa opção é melhor para coleções de documentos grandes.
   - Um documento por linha: a entrada é um ou mais arquivos. Cada linha em um arquivo é considerada um documento. Essa opção é melhor para documentos curtos, como publicações em mídias sociais. Cada linha deve terminar com uma alimentação de linha (LF, \n), um retorno de carro (CR, \r) ou ambos (CRLF, \r\n). Você não pode usar o separador de linha UTF-8 (u+2028) para finalizar uma linha.
10. Em Dados de saída, escolha Procurar no S3. Escolha o bucket ou pasta do Amazon S3 em que você deseja que o Amazon Comprehend grave os dados de saída produzidos pela análise.
11. (Opcional) Para criptografar o resultado de saída da sua tarefa, escolha Criptografia. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta:
    - Se você estiver usando uma chave associada à conta atual, escolha o alias ou ID da chave para o ID da chave KMS.
    - Se você estiver usando uma chave associada a uma conta diferente, insira o ARN do alias ou ID da chave em ID da chave KMS.

> **Note**
>
> Para obter mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte Serviço de gerenciamento de chaves (KMS).

12. Em Permissões de acesso, forneça um perfil do IAM que:

    - Conceda acesso de leitura ao local dos documentos de entrada do Amazon S3.
    - Conceda acesso de gravação ao local dos documentos de saída do Amazon S3.
    - Inclua uma política de confiança que permita a entidade principal do serviço do `comprehend.amazonaws.com` de assumir o perfil e obter suas permissões.

    Se você ainda não tem um perfil do IAM com essas permissões e uma política de confiança apropriada, escolha Criar um perfil do IAM.

13. Quando terminar de preencher o formulário, escolha Criar tarefa para criar e iniciar a tarefa de detecção de tópicos.

A nova tarefa aparece na lista de tarefas mostrando seu status no campo com essa finalidade. O campo pode ser `IN_PROGRESS` para uma tarefa em processamento, `COMPLETED` para uma que foi concluída com êxito e `FAILED` para uma tarefa com erro. Você pode clicar em uma tarefa para obter mais informações sobre ela, incluindo mensagens de erro.

Quando a tarefa é concluída, o Amazon Comprehend armazena os resultados da análise no local de saída do Amazon S3 especificado para a tarefa. Para obter uma descrição dos resultados da análise, consulte Detectar entidades de PII.

Análise em tempo real (API) de PII

O Amazon Comprehend fornece operações de API síncronas em tempo real para analisar informações de identificação pessoal (PII) em um documento.

Tópicos

- Localizando entidades de PII em tempo real (API)
- Rotular entidades de PII em tempo real (API)

Localizando entidades de PII em tempo real (API)

Para localizar PII em um único documento, você pode usar a operação Amazon Comprehend DetectPiiEntities. Seu texto de entrada pode incluir até 100 kilobytes de caracteres codificados em UTF-8. Os idiomas suportados incluem inglês e espanhol.

Localizar PII usando (CLI)

O exemplo a seguir usa a operação `DetectPiiEntities` com o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
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
```

O Amazon Comprehend responde com o seguinte:

```
{
      "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
          }
```

```
      ]
  }
```

Rotular entidades de PII em tempo real (API)

Você pode usar operações de API síncronas em tempo real para retornar os rótulos dos tipos de entidades de PII identificados. Para obter mais informações, consulte Rotular entidades de PII.

Rotular entidades PII (CLI)

O exemplo a seguir usa a operação `ContainsPiiEntities` com o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
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
```

O Amazon Comprehend responde com o seguinte:

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

Tarefas de análise assíncrona de PII (API)

Análise assíncrona de PII (API)

Você pode usar operações assíncronas de API para criar tarefas de análise a fim de localizar ou redigir entidades de PII. Para mais informações sobre os tipos de entidade de PII, consulte Detectar entidades de PII.

Tópicos

- Localizar entidades de PII com tarefas assíncronas (API)
- Redigir entidades de PII com tarefas assíncronas (API)

Localizar entidades de PII com tarefas assíncronas (API)

Execute uma tarefa em lotes assíncronos para localizar PII em uma coleção de documentos. Para executar o trabalho, faça o upload de seus documentos para o Amazon S3 e envie uma StartPiiEntitiesDetectionJobsolicitação.

Tópicos

- Antes de começar
- Parâmetros de entrada
- Métodos de tarefas assíncronos
- Formato do arquivo de saída
- Análise assíncrona usando o AWS Command Line Interface

Antes de começar

Antes de começar, garanta que você tenha:

- Buckets de entrada e saída: identifique os buckets do Amazon S3 que você deseja usar para entrada e saída. O buckets devem estar na mesma região que o API que você está chamando.
- Perfil de serviço do IAM: você deve ter um perfil de serviço do IAM com permissão para acessar seus buckets de entrada e saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Parâmetros de entrada

Em sua solicitação, inclua os seguintes parâmetros necessários:

- `InputDataConfig`— forneça uma InputDataConfigdefinição para sua solicitação, que inclua as propriedades de entrada para o trabalho. Para o parâmetro `S3Uri`, especifique a localização dos documentos de entrada no Amazon S3.
- `OutputDataConfig`— forneça uma OutputDataConfigdefinição para sua solicitação, que inclua as propriedades de saída do trabalho. Para o parâmetro `S3Uri`, especifique a localização do Amazon S3 em que o Amazon Comprehend grava os resultados de sua análise.
- `DataAccessRoleArn`— Forneça o Amazon Resource Name (ARN) de uma AWS Identity and Access Management função. Esse perfil deve conceder ao Amazon Comprehend acesso de leitura aos seus dados de entrada e acesso de gravação ao seu local de saída no Amazon S3. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.
- `Mode`: defina este parâmetro como `ONLY_OFFSETS`. Com essa configuração, a saída fornece os offsets de caracteres que localizam cada entidade de PII no texto de entrada. O resultado também inclui pontuações de confiança e tipos de entidades de PII.
- `LanguageCode`— Defina esse parâmetro como `en` ou`es`. O Amazon Comprehend oferece suporte à detecção de PII em texto em inglês ou espanhol.

Métodos de tarefas assíncronos

O `StartPiiEntitiesDetectionJob` retorna um ID da tarefa para que você possa monitorar seu andamento e recuperar seu status quando ele for concluído.

Para monitorar o progresso de um trabalho de análise, forneça o ID do trabalho para a DescribePiiEntitiesDetectionJoboperação. A resposta de `DescribePiiEntitiesDetectionJob` contém o campo `JobStatus` com o status atual da tarefa. Uma tarefa bem-sucedida passa pelos seguintes estados:

SUBMITTED -> IN_PROGRESS -> COMPLETED.

Depois que uma tarefa de análise for concluída (`JobStatus` para COMPLETED, FAILED, ou STOPPED), use `DescribePiiEntitiesDetectionJob` para obter a localização dos resultados. Se o status da tarefa for `COMPLETED`, a resposta incluirá um campo `OutputDataConfig` contendo um outro campo com a localização do arquivo de saída no Amazon S3.

Para obter detalhes adicionais sobre as etapas a serem seguidas para a análise assíncrona do Amazon Comprehend, consulte Processamento em lote assíncrono.

Formato do arquivo de saída

O arquivo de saída usa o nome do arquivo de entrada, com .out anexo no final. Ele contém os resultados da análise.

Veja a seguir um exemplo de arquivo de saída de uma tarefa de análise que detectou entidades de PII em documentos. O formato da entrada é um documento por linha.

```
{
  "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
 }
```

Veja a seguir um exemplo de saída de uma análise em que o formato da entrada é um documento por arquivo.

```
{
  "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"
}
```

Análise assíncrona usando o AWS Command Line Interface

O exemplo a seguir usa a operação `StartPiiEntitiesDetectionJob` com o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Para o parâmetro `cli-input-json`, você fornece um caminho para um arquivo JSON que contém os dados da solicitação, conforme exibido no exemplo a seguir.

```
{
  "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"
}
```

Você receberá uma resposta parecida com a seguinte se a solicitação para iniciar a tarefa de detecção de eventos for bem-sucedida:

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

Você pode usar a DescribeEventsDetectionJoboperação para obter o status de um trabalho existente. Você receberá uma resposta parecida com a seguinte se a solicitação para iniciar a tarefa de detecção de eventos for bem-sucedida:

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

Quando a tarefa for concluída com êxito, você receberá uma resposta semelhante a seguinte:

```
{
    "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)
    }
}
```

Redigir entidades de PII com tarefas assíncronas (API)

Para redigir as entidades de PII em seu texto, inicie uma tarefa em lotes assíncronos de tarefa. Para executar o trabalho, faça o upload de seus documentos para o Amazon S3 e envie uma StartPiiEntitiesDetectionJobsolicitação.

Tópicos

- Antes de começar
- Parâmetros de entrada
- Formato do arquivo de saída
- Redação de PII usando o AWS Command Line Interface

Antes de começar

Antes de começar, garanta que você tenha:

- Buckets de entrada e saída: identifique os buckets do Amazon S3 que você deseja usar para entrada e saída. O buckets devem estar na mesma região que o API que você está chamando.

- Perfil de serviço do IAM: você deve ter um perfil de serviço do IAM com permissão para acessar seus buckets de entrada e saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Parâmetros de entrada

Em sua solicitação, inclua os seguintes parâmetros necessários:

- `InputDataConfig`— forneça uma InputDataConfigdefinição para sua solicitação, que inclua as propriedades de entrada para o trabalho. Para o parâmetro `S3Uri`, especifique a localização dos documentos de entrada no Amazon S3.
- `OutputDataConfig`— forneça uma OutputDataConfigdefinição para sua solicitação, que inclua as propriedades de saída do trabalho. Para o parâmetro `S3Uri`, especifique a localização do Amazon S3 em que o Amazon Comprehend grava os resultados de sua análise.
- `DataAccessRoleArn`: forneça o nome do recurso da Amazon (ARN) de um perfil do AWS Identity and Access Management . Esse perfil deve conceder ao Amazon Comprehend acesso de leitura aos seus dados de entrada e acesso de gravação ao seu local de saída no Amazon S3. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.
- `Mode`: defina este parâmetro como `ONLY_REDACTION`. Com essa configuração, o Amazon Comprehend grava uma cópia dos seus documentos de entrada no local de saída no Amazon S3. Nessa cópia, cada entidade de PII é redigida.
- `RedactionConfig`— forneça uma RedactionConfigdefinição para sua solicitação, que inclua os parâmetros de configuração para a redação. Especifique os tipos de PII a serem redigidos e se cada entidade de PII será substituída pelo nome de seu tipo ou por um caractere de sua escolha:
  - Especifique os tipos de entidade de PII a serem redigidos na matriz de `PiiEntityTypes`. Para redigir todos os tipos de entidade, defina o valor da matriz como `["ALL"]`.
  - Para substituir cada entidade de PII por seu tipo, defina o parâmetro `MaskMode` como `REPLACE_WITH_PII_ENTITY_TYPE`. Por exemplo, com essa configuração, a entidade de PII “Jane Doe” é substituída por “[NAME]”.
  - Para substituir os caracteres em cada entidade de PII por um caractere de sua escolha, defina o parâmetro `MaskMode` como `MASK` e defina o parâmetro `MaskCharacter` como o caractere substituto. Forneça somente um único caractere. Os caracteres válidos são !, #, $, %, &, *, and @. Por exemplo, com essa configuração, a entidade de PII “Jane Doe” é substituída por "**** ***"

- `LanguageCode`— Defina esse parâmetro como `en` ou `es`. O Amazon Comprehend oferece suporte à detecção de PII em texto em inglês ou espanhol.

Formato do arquivo de saída

O exemplo a seguir mostra os arquivos de entrada e saída de uma tarefa de análise que redige PII. O formato da entrada é um documento por linha.

```
{
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
```

A tarefa de análise para redigir esse arquivo de entrada produz o seguinte arquivo de saída.

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

Redação de PII usando o AWS Command Line Interface

O exemplo a seguir usa a operação `StartPiiEntitiesDetectionJob` com o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

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

Para o parâmetro `cli-input-json`, você fornece um caminho para um arquivo JSON que contém os dados da solicitação, conforme exibido no exemplo a seguir.

```
{
```

```
    "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"]
    }
}
```

Você receberá uma resposta parecida com a seguinte se a solicitação para iniciar a tarefa de detecção de eventos for bem-sucedida:

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

Você pode usar a DescribeEventsDetectionJoboperação para obter o status de um trabalho existente.

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

Quando a tarefa for concluída com êxito, você receberá uma resposta semelhante a seguinte:

```
{
  "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)
   }
}
```

Processamento de documentos

O Amazon Comprehend é compatível com o processamento de documentos em uma etapa para classificação personalizada e reconhecimento personalizado de entidades. Por exemplo, você pode inserir uma combinação de documentos de texto simples e documentos semiestruturados (como documentos em PDF, documentos do Microsoft Word e imagens) em um trabalho de análise personalizada.

Para arquivos de entrada que exigem extração de texto, o Amazon Comprehend executa, automaticamente, a extração de texto antes de executar a análise. Para extrair o conteúdo do texto, o Amazon Comprehend usa um analisador interno para documentos semiestruturados nativos e usa o Amazon Textract para imagens e documentos digitalizados. APIs

O processamento de documentos do Amazon Comprehend está disponível em cada Amazon Comprehend, exceto na região Ásia-Pacífico (Tóquio) AWS GovCloud e (Oeste dos EUA) que oferecem suporte somente a modelos de texto simples para classificação personalizada. Regiões do compatíveis

Os tópicos a seguir fornecem detalhes sobre os tipos de documentos de entrada com os quais o Amazon Comprehend é compatível para análise personalizada.

Tópicos

- Entradas para análise personalizada em tempo real
- Entradas para análise personalizada assíncrona
- Configurar opções de extração de texto
- Práticas recomendadas para imagens

Entradas para análise personalizada em tempo real

A análise em tempo real usando modelos personalizados usa um único documento como entrada. Os tópicos a seguir descrevem os tipos de documentos de entrada que você pode usar.

Tópicos

- Documentos de texto simples
- Documentos semiestruturados

- Arquivos de imagem e arquivos PDF digitalizados
- Saída do Amazon Textract
- Tamanhos máximos de documentos para análise em tempo real
- Erros em documentos semiestruturados

Documentos de texto simples

Forneça os documentos de entrada como texto no formato UTF-8.

Documentos semiestruturados

Os documentos semiestruturados incluem documentos PDF nativos e documentos do Word.

Por padrão, a análise personalizada em tempo real usa o analisador do Amazon Comprehend para extrair o texto de arquivos Word e arquivos PDF digitais. Para arquivos PDF, você pode substituir esse padrão e usar o Amazon Textract para extrair o texto. Consulte Configurar opções de extração de texto.

Arquivos de imagem e arquivos PDF digitalizados

Os tipos de imagem compatíveis incluem JPEG, PNG e TIFF.

Por padrão, o reconhecimento personalizado de identidades usa a operação de API `DetectDocumentText` do Amazon Textract para extrair o texto de arquivos de imagem e arquivos PDF digitalizados. Como alternativa, você pode substituir esse padrão para usar a operação de API `AnalyzeDocument`. Consulte Configurar opções de extração de texto.

Saída do Amazon Textract

Você pode fornecer a saída JSON da API `DetectDocumentText` ou da API `AnalyzeDocument` do Amazon Textract como entrada para as operações de API em tempo real para classificação personalizada e reconhecimento personalizado de entidades. O Amazon Comprehend oferece suporte a esse tipo de entrada para as operações de API em tempo real, mas não para o console.

Tamanhos máximos de documentos para análise em tempo real

Para todos os tipos de documentos de entrada, o máximo do arquivo de entrada é de uma página, com no máximo 10.000 caracteres.

A tabela a seguir mostra os tamanhos máximos de arquivo dos documentos de entrada.

Tipo de arquivo	Tamanho máximo (API)	Tamanho máximo (console)
Documentos de texto em UTF-8	10 KB	10 KB
Documentos em PDF	10 MB	5 MB
Documentos do Word	10 MB	1 MB
Arquivos de imagem	10 MB	5 MB
Arquivos de saída do Textract	1 MB	n/a

Erros em documentos semiestruturados

A operação ClassifyDocumentou DetectEntitiesAPI pode encontrar erros no nível do documento ou da página ao extrair texto de um documento semiestruturado ou de um arquivo de imagem.

Erros no nível da página

Se a operação da DetectEntitiesAPI ClassifyDocumentou encontrar erros ao processar uma página no documento de entrada, a resposta da API incluirá uma entrada na lista de erros para cada erro.

O `ErrorCode` na lista de erros contém um dos seguintes valores:

- TEXTRACT_BAD_PAGE: o Amazon Textract não consegue ler a página. Para obter mais informações sobre limites de página no Amazon Textract, consulte Cotas de página no Amazon Textract.
- TEXTRACT_PROVISIONED_THROUGHPUT_EXCEEDED: o número de solicitações excedeu seu limite de throughput. Para obter mais informações sobre cotas de throughput no Amazon Textract, consulte Cotas padrão no Amazon Textract.
- PAGE_CHARACTERS_EXCEEDED: muitos caracteres de texto na página (máximo de 10.000 caracteres).
- PAGE_SIZE_EXCEEDED: o tamanho máximo da página é 10 MB.
- INTERNAL_SERVER_ERROR: a solicitação encontrou um problema de serviço. Tente a solicitação da API novamente.

Erros no nível de documento

Se a operação ClassifyDocumentou DetectEntitiesAPI detectar um erro em nível de documento em seu documento de entrada, a API retornará uma `InvalidRequestException` resposta de erro.

Na resposta de erro, o valor no campo Reason irá conter o valor `INVALID_DOCUMENT`.

O campo Detail contém um dos seguintes valores:

- DOCUMENT_SIZE_EXCEEDED: o tamanho do documento é muito grande. Verifique o tamanho do seu arquivo e reenvie a solicitação.
- UNSUPPORTED_DOC_TYPE: o tipo de documento não é compatível. Verifique o tipo do seu arquivo e reenvie a solicitação.
- PAGE_LIMIT_EXCEEDED: muitas páginas no documento. Verifique o número de páginas do seu arquivo e reenvie a solicitação.
- TEXTRACT_ACCESS_DENIED_EXCEPTION: acesso negado ao Amazon Textract. Verifique se sua conta tem permissão para usar as operações do Amazon Textract DetectDocumentTexte AnalyzeDocumentda API e reenvie a solicitação.

Entradas para análise personalizada assíncrona

Você pode inserir vários documentos em um trabalho assíncrono de análise personalizada. Os tópicos a seguir descrevem os tipos de documentos de entrada que você pode usar. O tamanho máximo de arquivo muda de acordo com o tipo de documento de entrada.

Tópicos

- Documentos de texto simples
- Documentos semiestruturados
- Arquivos de imagem e arquivos PDF digitalizados
- Arquivos JSON de saída do Amazon Textract

Documentos de texto simples

Forneça todos os documentos de entrada de texto sem formatação como texto no formato UTF-8. A tabela a seguir lista os tamanhos máximos de arquivo e outras diretrizes.

> **Note**
>
> Esses limites se aplicam quando todos os arquivos de entrada forem texto simples.

Descrição	Cota/diretriz
Tamanho máximo de arquivo para um documento por formato de arquivo (classificação personalizada)	1 byte a 10 MB
Tamanho do documento (reconhecimento personalizado de entidades)	1 byte a 1 MB
Número máximo de arquivos, um documento por arquivo	1.000.000
Número máximo de linhas, um documento por linha (para todos os arquivos solicitados)	1.000.000
Tamanho do corpo do documento (todos os documentos em texto simples combinados)	1 byte a 5 GB

Documentos semiestruturados

Os documentos semiestruturados incluem documentos PDF nativos e documentos do Word.

A tabela a seguir lista os tamanhos máximos de arquivo e outras diretrizes.

Descrição	Cota/diretriz
Tamanho do documento (PDF)	1 byte a 50 MB
Tamanho do documento (Docx)	1 byte a 5 MB
Número máximo de perfis	500
Número máximo de páginas para um arquivo PDF ou Docx	100

Descrição	Cota/diretriz
Tamanho do corpus do documento após a extração do texto (texto simples, todos os arquivos combinados)	1 byte a 5 GB

Por padrão, a análise personalizada usa o analisador do Amazon Comprehend para extrair o texto de arquivos Word e arquivos PDF digitais. Para arquivos PDF, você pode substituir esse padrão e usar o Amazon Textract para extrair o texto. Consulte Configurar opções de extração de texto.

Arquivos de imagem e arquivos PDF digitalizados

A análise personalizada é compatível com imagens JPEG, PNG e TIFF.

A tabela a seguir lista os tamanhos máximos de arquivo para imagens. Os arquivos PDF digitalizados estão sujeitos aos mesmos tamanhos máximos dos arquivos PDF nativos.

Descrição	Cota/diretriz
Tamanho da imagem (JPG ou PNG)	1 byte a 10 MB
Tamanho da imagem (TIFF)	1 byte a 10 MB. Máximo de uma página.

Para obter informações adicionais sobre conformidade, consulte Práticas recomendadas para imagens.

Por padrão, o Amazon Comprehend usa a operação de API `DetectDocumentText` do Amazon Textract para extrair o texto de arquivos de imagem e arquivos PDF digitalizados. Como alternativa, você pode substituir esse padrão para usar a operação de API `AnalyzeDocument`. Consulte Configurar opções de extração de texto.

Arquivos JSON de saída do Amazon Textract

Para reconhecimento de entidades personalizadas, mas não para classificação personalizada, você pode fornecer o arquivo de saída da operação de API `AnalyzeDocument` do Amazon Textract como entrada para trabalhos de análise.

Configurar opções de extração de texto

Por padrão, o Amazon Comprehend executa as seguintes ações para extrair texto de um arquivo, com base no tipo de arquivo de entrada:

- Arquivos do Word: o analisador do Amazon Comprehend extrai o texto.
- Arquivos em PDF digitais: o analisador do Amazon Comprehend extrai o texto.
- Arquivos de imagem e em PDF digitalizados: O Amazon Comprehend usa a API do Amazon Textract `DetectDocumentText` para extrair o texto.

Para arquivos de imagem e arquivos em PDF, você pode usar o parâmetro `DocumentReaderConfig` para substituir essas ações padrão de extração. Esse parâmetro está disponível quando você usa o console ou a API do Amazon Comprehend para análises personalizadas assíncronas ou em tempo real.

O parâmetro `DocumentReaderConfig` contém três campos:

- DocumentReadMode— Defina como `SERVICE_DEFAULT` para que o Amazon Comprehend execute as ações padrão.

  Define `FORCE_DOCUMENT_READ_ACTION` para usar o Amazon Textract para analisar arquivos em PDF digitais.
- DocumentReadAction— Define a API Amazon Textract (DetectDocumentText ou AnalyzeDocument) para usar quando o Amazon Comprehend usa o Amazon Textract para extração de texto.
- FeatureTypes— Se você DocumentReadActiondefinir o uso da operação de AnalyzeDocument API, poderá adicionar uma ou ambas `FeatureTypes` (TABELAS, FORMULÁRIOS). Esses recursos fornecem informações adicionais sobre as tabelas e formulários no documento. Para obter mais informações sobre esses recursos, consulte Objetos de resposta da análise de documentos do Amazon Textract.

Os exemplos a seguir mostram como configurar `DocumentReaderConfig` para casos de uso específico:

1. Use o Amazon Textract para todos os arquivos em PDF.
   a. DocumentReadMode: defina como `FORCE_DOCUMENT_READ_ACTION`.

   b. DocumentReadAction: defina como `TEXTRACT_DETECT_DOCUMENT_TEXT`.
   c. FeatureTypes— Não é necessário.
2. Use a API `AnalyzeDocument` do Amazon Textract para todos os arquivos em PDF e de imagem.
   a. DocumentReadMode: defina como `FORCE_DOCUMENT_READ_ACTION`.
   b. DocumentReadAction: defina como `TEXTRACT_ANALYZE_DOCUMENT`.
   c. FeatureTypes— Defina `TABLES` como `FORMS` ou ambos os recursos.
3. Use a API `AnalyzeDocument` do Amazon Textract para arquivos em PDF digitalizados e todos os arquivos de imagem.
   a. DocumentReadMode: defina como `SERVICE_DEFAULT`.
   b. DocumentReadAction: defina como `TEXTRACT_ANALYZE_DOCUMENT`.
   c. FeatureTypes— Defina `TABLES` como `FORMS` ou ambos os recursos.

Para obter mais informações sobre as opções do Amazon Textract, consulte. DocumentReaderConfig

Práticas recomendadas para imagens

Ao usar arquivos de imagem para classificação personalizada ou reconhecimento personalizado de entidades, utilize as diretrizes a seguir para obter os melhores resultados:

- Forneça uma imagem de alta qualidade, de preferência com ao menos 150 DPI.
- Se o arquivo de imagem usar um dos formatos compatíveis (TIFF, JPEG ou PNG), não converta nem reduza a resolução do arquivo antes de enviá-lo para o Amazon S3.

Para obter os melhores resultados ao extrair texto de tabelas em documentos, siga estas práticas:

- As tabelas em seu documento estão separadas visualmente dos elementos ao redor da página. Por exemplo, a tabela não está sobreposta a uma imagem ou padrão complexo.
- O texto dentro da tabela está na vertical. Por exemplo, o texto não está girado em relação a outro texto na página.

Ao extrair texto de tabelas, você poderá obter resultados inconsistentes nos seguintes casos:

- Células mescladas da tabela abrangendo várias colunas.

- Tabelas com células, linhas ou colunas diferentes de outras partes da mesma tabela.

Classificação personalizada

Use a classificação personalizada para organizar seus documentos em categorias (classes) definidas por você. A classificação personalizada é um processo em duas etapas. Primeiro, você treina um modelo de classificação personalizado (também chamado de classificador) para reconhecer as classes que são do seu interesse. Em seguida, você usa seu modelo para classificar qualquer número de conjuntos de documentos.

Por exemplo, você pode categorizar o conteúdo das solicitações de suporte a fim de poder encaminhar a solicitação para a equipe de suporte adequada. Ou você pode categorizar os e-mails recebidos dos clientes para fornecer orientação com base no tipo de solicitação do cliente. Você pode combinar o Amazon Comprehend com o Amazon Transcribe para converter fala em texto e depois classificar as solicitações provenientes de chamadas telefônicas de suporte.

Você pode executar a classificação personalizada em um único documento de forma síncrona (em tempo real) ou iniciar uma tarefa assíncrona para classificar um conjunto de documentos. Você pode ter vários classificadores personalizados em sua conta, cada um treinado usando dados diferentes. A classificação personalizada é compatível com uma variedade de tipos de documentos de entrada, como texto sem formatação, PDF, Word e imagens.

Ao enviar uma tarefa de classificação, você escolhe o modelo de classificador a ser usado, com base no tipo de documentos que precisa analisar. Por exemplo, para analisar documentos de texto sem formatação, você obtém os resultados mais precisos usando um modelo treinado com documentos de texto sem formatação. Para analisar documentos semiestruturados (como PDF, Word, imagens, saída do Amazon Textract ou arquivos digitalizados), você obtém os resultados mais precisos usando um modelo treinado com documentos originais.

Tópicos

- Preparar dados de treinamento do classificador
- Modelos de classificação de treinamento
- Executar análises em tempo real
- Executar tarefas assíncronas

Preparar dados de treinamento do classificador

Para classificação personalizada, você treina o modelo no modo multiclasse ou no modo multirrótulos. O modo multiclasse associa uma única classe a cada documento. O modo multirrótulos associa uma ou mais classes a cada documento. Os formatos de arquivo de entrada são diferentes para cada modo, então escolha o modo a ser usado antes de criar os dados de treinamento.

> **Note**
>
> O console do Amazon Comprehend se refere ao modo multiclasse como modo de rótulo único.

A classificação personalizada é compatível com modelos que você treina com documentos sem formatação e modelos treinados com documentos originais (como PDF, Word ou imagens). Para mais informações sobre modelos de classificadores e seus tipos de documentos compatíveis, consulte Modelos de classificação de treinamento.

Preparar dados para treinar um modelo de classificador personalizado:

1. Identifique as classes que você deseja que esse classificador analise. Decida qual modo usar (multiclasse ou multirrótulo).
2. Decida o tipo de modelo do classificador, com base no modelo ter o fim de analisar documentos de texto simples ou documentos semiestruturados.
3. Reúna exemplos de documentos para cada uma das classes. Para obter os requisitos mínimos de treinamento, consulteCotas gerais para classificação de documentos.
4. Para um modelo de texto sem formatação, escolha o formato de arquivo de treinamento a ser usado (arquivo CSV ou arquivo de manifesto aumentado). Para treinar um modelo de documento nativo, você sempre usa um arquivo CSV.

Tópicos

- Formatos de arquivo de treinamento do classificador
- Modo multiclasse
- Modo multirrótulo

Formatos de arquivo de treinamento do classificador

Para um modelo de texto sem formatação, você pode fornecer dados de treinamento do classificador como um arquivo CSV ou como um arquivo de manifesto aumentado que você cria usando o AI Ground Truth. SageMaker O arquivo CSV ou arquivo de manifesto aumentado inclui o texto de cada documento de treinamento e seus rótulos associados.

Para um modelo de documento nativo, você fornece dados do classificador de treinamento como um arquivo CSV. O arquivo CSV inclui o nome do arquivo de cada documento de treinamento e seus rótulos associados. Você inclui os documentos de treinamento na pasta de entrada do Amazon S3 para a tarefa de treinamento.

Arquivos CSV

Você fornece dados de treinamento rotulados como texto codificado em UTF-8 em um arquivo CSV. Não inclua uma linha de cabeçalho. Adicionar uma linha de cabeçalho em seu arquivo pode causar erros runtime.

Para cada linha no arquivo CSV, a primeira coluna contém um ou mais rótulos de classe. Um rótulo de classe pode ser qualquer string UTF-8 válida. Recomendamos usar nomes de classe claros que não se sobreponham em significado. O nome pode incluir espaço em branco e consistir em várias palavras conectadas por sublinhados ou hífens.

Não deixe nenhum caractere de espaço antes ou depois das vírgulas que separam os valores em uma linha.

O conteúdo exato do arquivo CSV depende do modo classificador e do tipo de dados de treinamento. Para mais detalhes, consulte as seções em Modo multiclasse e Modo multirrótulo.

Arquivo de manifesto aumentado

Um arquivo de manifesto aumentado é um conjunto de dados rotulado que você cria usando o AI Ground SageMaker Truth. O Ground Truth é um serviço de rotulagem de dados que ajuda você ou seus funcionários a criar conjuntos de dados de treinamento para modelos de machine learning.

Para obter mais informações sobre o Ground Truth e os resultados que ele produz, consulte Use o SageMaker AI Ground Truth to Label Data no Amazon SageMaker AI Developer Guide.

Os arquivos de manifesto aumentado estão no formato de linhas JSON. Nesses arquivos, cada linha é um objeto JSON completo que contém um documento de treinamento e seus rótulos associados.

O conteúdo exato de cada linha depende do modo classificador. Para mais detalhes, consulte as seções em Modo multiclasse e Modo multirrótulo.

Ao fornecer seus dados de treinamento ao Amazon Comprehend, você especifica um ou mais nomes de atributos do rótulo. A quantidade de nomes de atributos que você especifica depende se seu arquivo de manifesto aumentado é a saída de uma única tarefa de rotulagem ou de uma tarefa de rotulagem em cadeia.

Se seu arquivo for a saída de uma tarefa de rotulagem única, especifique o nome do atributo de rótulo único da tarefa do Ground Truth.

Se seu arquivo for a saída de um trabalho de rotulagem em cadeia, especifique o nome de atributo de rótulo etiqueta para um ou mais trabalhos na cadeia. Cada nome de atributo de rótulo fornece as anotações de uma tarefa individual. Você pode especificar até 5 desses atributos para arquivos de manifesto aumentados a partir de tarefas de rotulagem encadeadas.

Para obter mais informações sobre trabalhos de etiquetagem em cadeia e para obter exemplos dos resultados que eles produzem, consulte Encadeamento de trabalhos de etiquetagem no Amazon SageMaker AI Developer Guide.

Modo multiclasse

No modo multiclasse, a classificação atribui uma classe para cada documento. As classes individuais são mutuamente exclusivas. Por exemplo, você pode classificar um filme como comédia ou ficção científica, mas não ambos.

> **Note**
>
> O console do Amazon Comprehend se refere ao modo multiclasse como modo de rótulo único.

Tópicos

- Modelos de texto sem formatação
- Modelos de documento originais

Modelos de texto sem formatação

Para treinar um modelo de texto simples, você pode fornecer dados de treinamento rotulados como um arquivo CSV ou como um arquivo de manifesto aumentado do AI Ground Truth. SageMaker

Arquivo CSV

Para informações gerais sobre como usar arquivos CSV para classificadores de treinamento, consulte Arquivos CSV.

Forneça os dados de treinamento como um arquivo CSV de duas colunas. Para cada linha, a primeira coluna contém o valor do rótulo da classe. A segunda coluna contém um exemplo de documento de texto para essa classe. Cada linha deve terminar com caracteres \n ou \r\n.

O exemplo a seguir mostra um arquivo CSV contendo três documentos.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS,Text of document 3
```

O exemplo a seguir mostra uma linha de um arquivo CSV que treina um classificador personalizado para detectar se uma mensagem de e-mail é spam:

```
SPAM,"Paulo, your $1000 award is waiting for you! Claim it while you still can at
 http://example.com."
```

Arquivo de manifesto aumentado

Para informações gerais sobre o uso de arquivos de manifesto aumentados para treinar classificadores, consulte Arquivo de manifesto aumentado.

Para documentos de texto sem formatação, cada linha do arquivo de manifesto aumentado é um objeto JSON completo contendo um documento de treinamento, um único nome de classe e outros metadados do Ground Truth. O exemplo a seguir é um arquivo de manifesto aumentado para treinar um classificador personalizado a fim de reconhecer mensagens de e-mail de spam:

```
{"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"}}
```

O exemplo a seguir mostra um objeto JSON do arquivo de manifesto aumentado, formatado para facilitar a leitura:

```
{
   "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"
   }
}
```

Neste exemplo, o atributo `source` fornece o texto do documento de treinamento e o atributo `MultiClassJob` atribui o índice de uma classe de uma lista de classificação. O atributo `job-name` é o nome que você definiu para a tarefa de rotulagem no Ground Truth.

Ao iniciar a tarefa de treinamento de classificadores no Amazon Comprehend, você especifica o mesmo nome da tarefa de rotulagem.

Modelos de documento originais

Um modelo de documento nativo é um modelo que você treina com documentos nativos (como PDF, DOCX e imagens). Você fornece os dados de treinamento como um arquivo CSV.

Arquivo CSV

Para informações gerais sobre como usar arquivos CSV para classificadores de treinamento, consulte Arquivos CSV.

Forneça os dados de treinamento como um arquivo CSV de três colunas. Para cada linha, a primeira coluna contém o valor do rótulo da classe. A segunda coluna contém o nome do arquivo de um documento de exemplo para essa classe. A terceira coluna contém o número da página. O número da página é opcional se o documento de exemplo for uma imagem.

O exemplo a seguir mostra um arquivo CSV que faz referência a três documentos de entrada.

```
CLASS,input-doc-1.pdf,3
CLASS,input-doc-2.docx,1
CLASS,input-doc-3.png
```

O exemplo a seguir mostra uma linha de um arquivo CSV que treina um classificador personalizado para detectar se uma mensagem de e-mail é spam. A página 2 do arquivo PDF contém o exemplo de spam.

```
SPAM,email-content-3.pdf,2
```

Modo multirrótulo

No modo multirrótulo, as classes individuais representam categorias diferentes que não são mutuamente exclusivas. A classificação multirrótulo atribui uma ou mais classes a cada documento. Por exemplo, você pode classificar um filme como documentário e outro filme como ficção científica, ação e comédia.

Para treinamento, o modo multirrótulo suporta até 1 milhão de exemplos contendo até 100 classes exclusivas.

Tópicos

- Modelos de texto sem formatação
- Modelos de documento originais

Modelos de texto sem formatação

Para treinar um modelo de texto simples, você pode fornecer dados de treinamento rotulados como um arquivo CSV ou como um arquivo de manifesto aumentado do AI Ground Truth. SageMaker

Arquivo CSV

Para informações gerais sobre como usar arquivos CSV para classificadores de treinamento, consulte Arquivos CSV.

Forneça os dados de treinamento como um arquivo CSV de duas colunas. Para cada linha, a primeira coluna contém os valores do rótulo da classe e a segunda coluna contém um exemplo de documento de texto para essas classes. Para inserir mais de uma classe na primeira coluna, use um delimitador (como |) entre cada classe.

```
CLASS,Text of document 1
CLASS,Text of document 2
CLASS|CLASS|CLASS,Text of document 3
```

O exemplo a seguir mostra uma linha de um arquivo CSV que treina um classificador personalizado para detectar gêneros em resumos de filmes:

```
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?"
```

O delimitador padrão entre os nomes das classes é uma barra vertical, também chamada de pipe (|). No entanto, é possível usar um caractere diferente como um delimitador. O delimitador deve ser diferente de todos os caracteres nos nomes de suas classes. Por exemplo, se suas classes são CLASSE_1, CLASSE_2 e CLASSE_3, o sublinhado (_) faz parte do nome da classe. Portanto, não use um sublinhado como delimitador para separar os nomes das classes.

Arquivo de manifesto aumentado

Para informações gerais sobre o uso de arquivos de manifesto aumentados para treinar classificadores, consulte Arquivo de manifesto aumentado.

Para documentos de texto sem formatação, cada linha do arquivo de manifesto aumentado é um objeto JSON completo. Ele contém um documento de treinamento, nomes de classes e outros metadados do Ground Truth. O exemplo a seguir é um arquivo de manifesto aumentado para treinar um classificador personalizado a fim de detectar gêneros em resumos de filmes:

```
{"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"}}
```

O exemplo a seguir mostra um objeto JSON do arquivo de manifesto aumentado, formatado para facilitar a leitura:

```
{
      "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"
      }
```

```
}
```

Neste exemplo, o atributo `source` fornece o texto do documento de treinamento e o atributo `MultiLabelJob` atribui os índices de várias classes de uma lista de classificação. O nome da tarefa nos metadados do `MultiLabelJob` é o nome que você definiu para a tarefa de rotulagem no Ground Truth.

Modelos de documento originais

Um modelo de documento nativo é um modelo que você treina com documentos nativos (como PDF, DOCX e arquivos de imagem). Você fornece dados de treinamento rotulados como um arquivo CSV.

Arquivo CSV

Para informações gerais sobre como usar arquivos CSV para classificadores de treinamento, consulte Arquivos CSV.

Forneça os dados de treinamento como um arquivo CSV de três colunas. Para cada linha, a primeira coluna contém os valores do rótulo da classe. A segunda coluna contém o nome do arquivo de um documento de exemplo para essas classes. A terceira coluna contém o número da página. O número da página é opcional se o documento de exemplo for uma imagem.

Para inserir mais de uma classe na primeira coluna, use um delimitador (como |) entre cada classe.

```
CLASS,input-doc-1.pdf,3
CLASS,input-doc-2.docx,1
CLASS|CLASS|CLASS,input-doc-3.png,2
```

O exemplo a seguir mostra uma linha de um arquivo CSV que treina um classificador personalizado para detectar gêneros em resumos de filmes. A página 2 do arquivo PDF contém o exemplo de um comedy/teen filme.

```
COMEDY|TEEN,movie-summary-1.pdf,2
```

O delimitador padrão entre os nomes das classes é uma barra vertical, também chamada de pipe (|). No entanto, é possível usar um caractere diferente como um delimitador. O delimitador deve ser diferente de todos os caracteres nos nomes de suas classes. Por exemplo, se suas classes são CLASSE_1, CLASSE_2 e CLASSE_3, o sublinhado (_) faz parte do nome da classe. Portanto, não use um sublinhado como delimitador para separar os nomes das classes.

Modelos de classificação de treinamento

Para treinar um modelo para classificação personalizada, você define as categorias e fornece documentos de exemplo para treinar o modelo personalizado. Você treina o modelo no modo multiclasse ou multirrótulo. O modo multiclasse associa uma única classe a cada documento. O modo multirrótulos associa uma ou mais classes a cada documento.

A classificação personalizada é compatível com dois tipos de modelos de classificadores: modelos de texto sem formatação e modelos de documentos nativos. Um modelo de texto sem formatação classifica documentos com base em seu conteúdo de texto. Um modelo de documento nativo também classifica documentos com base no conteúdo do texto. Um modelo de documento nativo também pode usar sinais adicionais, como do layout do documento. Você treina um modelo de documento nativo com documentos nativos para que o modelo aprenda as informações do layout.

Os modelos de texto sem formatação têm as seguintes características:

- Você treina o modelo usando documentos de texto codificados em UTF-8.
- Você pode treinar o modelo usando documentos em um dos seguintes idiomas: inglês, espanhol, alemão, italiano, francês ou português.
- Todos os documentos de treinamento de um determinado classificador devem usar o mesmo idioma.
- Os documentos de treinamento são em texto sem formatação, portanto, não há custos adicionais pela extração de texto.

Os modelos de documentos nativos têm as seguintes características:

- Você treina o modelo usando documentos semiestruturados, que incluem os seguintes tipos de documentos:
  - Documentos PDF digitais e digitalizados.
  - Documentos do Word (DOCX).
  - Imagens: arquivos JPG, arquivos PNG e arquivos TIFF de página única.
  - Arquivos de saída JSON da API Textract.
- Você treina o modelo usando documentos em inglês.
- Documentos de treinamento incluindo arquivos de documentos digitalizados implicam em cobranças adicionais pela extração de texto. Consulte a página de Valores do Amazon Comprehend para detalhes.

Você pode classificar qualquer um dos tipos de documentos compatíveis usando qualquer tipo de modelo. No entanto, para obter resultados mais precisos, recomendamos o uso de um modelo de texto sem formatação para classificar documentos de texto sem formatação e um modelo de documento nativo para classificar documentos semiestruturados.

Tópicos

- Treine classificadores personalizados (console)
- Treine classificadores personalizados (API)
- Teste os dados de treinamento
- Saída do treinamento do classificador
- Métricas de classificador personalizados

Treine classificadores personalizados (console)

Você pode criar e treinar um classificador personalizado usando o console e, em seguida, usar o classificador personalizado para analisar seus documentos.

Para treinar um classificador personalizado, você precisa de um conjunto de documentos de treinamento. Você rotula esses documentos com as categorias que deseja que o classificador de documentos reconheça. Para informações sobre como preparar seus documentos de treinamento, consulte Preparar dados de treinamento do classificador.

Criar e treinar um modelo de classificador de documentos

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Personalização e, em seguida, Classificação personalizada.
3. Escolha Criar novo modelo.
4. Em Configurações do modelo, insira um nome de modelo para o classificador. O nome deve ser exclusivo na sua conta e na sua região atual.

   (Opcional) Insira um nome de versão. O nome deve ser exclusivo na sua conta e na sua região atual.
5. Selecione o idioma dos documentos de treinamento. Para ver os idiomas com os quais os classificadores são compatíveis, consulte Modelos de classificação de treinamento.

6. (Opcional) Se você quiser criptografar os dados no volume de armazenamento enquanto o Amazon Comprehend processa sua tarefa de treinamento, escolha Criptografia classificadora. Em seguida, selecione se deseja usar uma chave KMS associada à sua conta atual ou uma de outra conta.

   - Se estiver usando uma chave associada à conta atual, escolha o ID da chave para o ID da chave KMS.
   - Se estiver usando uma chave associada a uma conta diferente, insira o ARN do ID da chave em ARN da chave KMS.

   > **Note**
   >
   > Para mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte o AWS Key Management Service (AWS KMS).

7. Em Especificações de dados, escolha o Tipo de modelo de treinamento a ser usado.

   - Documentos de texto sem formatação: escolha essa opção para criar um modelo de texto sem formatação. Treine o modelo usando documentos de texto sem formatação.
   - Documentos nativos: escolha essa opção para criar um modelo de documento nativo. Treine o modelo usando documentos nativos (PDF, Word, imagens).

8. Escolha o Formato de dados dos seus dados de treinamento. Para mais informações sobre o formato, consulte Formatos de arquivo de treinamento do classificador.

   - Arquivo CSV: escolha essa opção se seus dados de treinamento usarem o formato de arquivo CSV.
   - Manifesto aumentado: escolha essa opção se você usou o Ground Truth para criar arquivos de manifesto aumentados para seus dados de treinamento. Esse formato estará disponível se você escolher Documentos de texto sem formatação como o tipo de modelo de treinamento.

9. Escolha o Modo classificador a ser usado.

   - Modo de rótulo único: escolha esse modo se as categorias que você está atribuindo aos documentos forem mutuamente exclusivas e você estiver treinando seu classificador para atribuir um rótulo a cada documento. Na API da Amazon Comprehend, o modo de rótulo único é conhecido como modo multiclasse.

- Modo multirrótulo: escolha esse modo se várias categorias puderem ser aplicadas a um documento ao mesmo tempo e você estiver treinando seu classificador para atribuir um ou mais rótulos a cada documento.

10. Se você escolher o Modo multirrótulo, poderá selecionar o Delimitador para rótulos. Use esse caractere delimitador para separar rótulos quando houver várias classes para um documento de treinamento. O delimitador padrão é o caractere pipe.
11. (Opcional) Se escolher Manifesto aumentado como formato de dados, poderá inserir até cinco arquivos de manifesto aumentados. Cada arquivo de manifesto aumentado contém um conjunto de dados de treinamento ou um conjunto de dados de teste. Você deve fornecer pelo menos um conjunto de dados de treinamento. Os conjuntos de dados de teste são opcionais. Use as seguintes etapas para configurar os arquivos de manifesto aumentados:

    a. Em Conjunto de dados de treinamento e teste, expanda o painel Localização de entrada.
    b. Em Tipo de conjunto de dados, escolha Dados de treinamento ou Dados de teste.
    c. Para a localização S3 do arquivo de manifesto aumentado do SageMaker AI Ground Truth, insira a localização do bucket Amazon S3 que contém o arquivo de manifesto ou navegue até ele escolhendo Browse S3. O perfil do IAM que você está usando para obter permissões a fim de acessar a tarefa de treinamento deve ter permissões para leitura no bucket do S3.
    d. Para os Nomes dos atributos, insira o nome do atributo que contém suas anotações. Se o arquivo contiver anotações de várias tarefas de rotulagem em cadeia, adicione um atributo para cada tarefa.
    e. Para adicionar outro local de entrada, escolha Adicionar local de entrada e configure o próximo local.

12. (Opcional) Se escolheu o Arquivo CSV como formato de dados, use as etapas a seguir para configurar o conjunto de dados de treinamento e o conjunto de dados de teste opcional:

    a. Em Conjunto de dados de treinamento, insira a localização do bucket do Amazon S3 que contém seu arquivo CSV de dados de treinamento ou navegue até ele escolhendo Procurar no S3. O perfil do IAM que você está usando para obter permissões a fim de acessar a tarefa de treinamento deve ter permissões para leitura no bucket do S3.

        (Opcional) Se escolheu Documentos nativos como o tipo de modelo de treinamento, você também fornecerá a URL da pasta do Amazon S3 contendo os arquivos de exemplo de treinamento.
    b. Em Testar conjunto de dados, selecione se você está fornecendo dados extras para o Amazon Comprehend testar o modelo treinado.

- Divisão automática: a divisão automática seleciona automaticamente 10% dos seus dados de treinamento a fim de reservar para uso como dados de teste.
- (Opcional) Fornecido pelo cliente: insira a URL do arquivo CSV de dados de teste no Amazon S3. Você também pode navegar até localização do arquivo no Amazon S3 e escolher Selecionar pasta.

  (Opcional) Se escolher Documentos nativos como o tipo de modelo de treinamento, você também fornecerá a URL da pasta do Amazon S3 contendo os arquivos de teste.

13. (Opcional) Para Modo de leitura do documento, você pode substituir as ações padrão de extração de texto. Essa opção não é necessária para modelos de texto sem formatação, já que se aplica à extração de texto para documentos digitalizados. Para obter mais informações, consulte Configurar opções de extração de texto.
14. (Opcional para modelos de texto sem formatação) Para Dados de saída, insira a localização de um bucket do Amazon S3 para salvar os dados de saída do treinamento, como a matriz de confusão. Para obter mais informações, consulte Matriz de confusão.

    (Opcional) Se criptografar o resultado de saída da sua tarefa de treinamento, escolha Criptografia. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta.

    - Se estiver usando uma chave associada à conta atual, escolha o alias da chave para o ID da chave KMS.
    - Se você estiver usando uma chave associada a uma conta diferente, insira o ARN do alias ou ID da chave em ID da chave KMS.
15. Para o Perfil do IAM, selecione Escolher um perfil do IAM existente e, em seguida, escolha um perfil do IAM existente com permissões de leitura para o bucket do S3 contendo seus documentos de treinamento. O perfil deve ter uma política de confiança que comece com `comprehend.amazonaws.com` para ser válida.

    Se ainda não tem um perfil do IAM com essas permissões, escolha Criar um perfil do IAM. Escolha as permissões de acesso para conceder esse perfil e, depois, escolha um sufixo de nome para diferenciar o perfil dos perfis do IAM em sua conta.

> **Note**
>
> Para documentos de entrada criptografados, o perfil do IAM usada também deve ter uma permissão do `kms:Decrypt`. Para obter mais informações, consulte Permissões necessárias para usar a criptografia KMS.

16. (Opcional) Para lançar seus recursos no Amazon Comprehend a partir de uma VPC, insira o ID da VPC em VPC ou escolha o ID na lista suspensa.

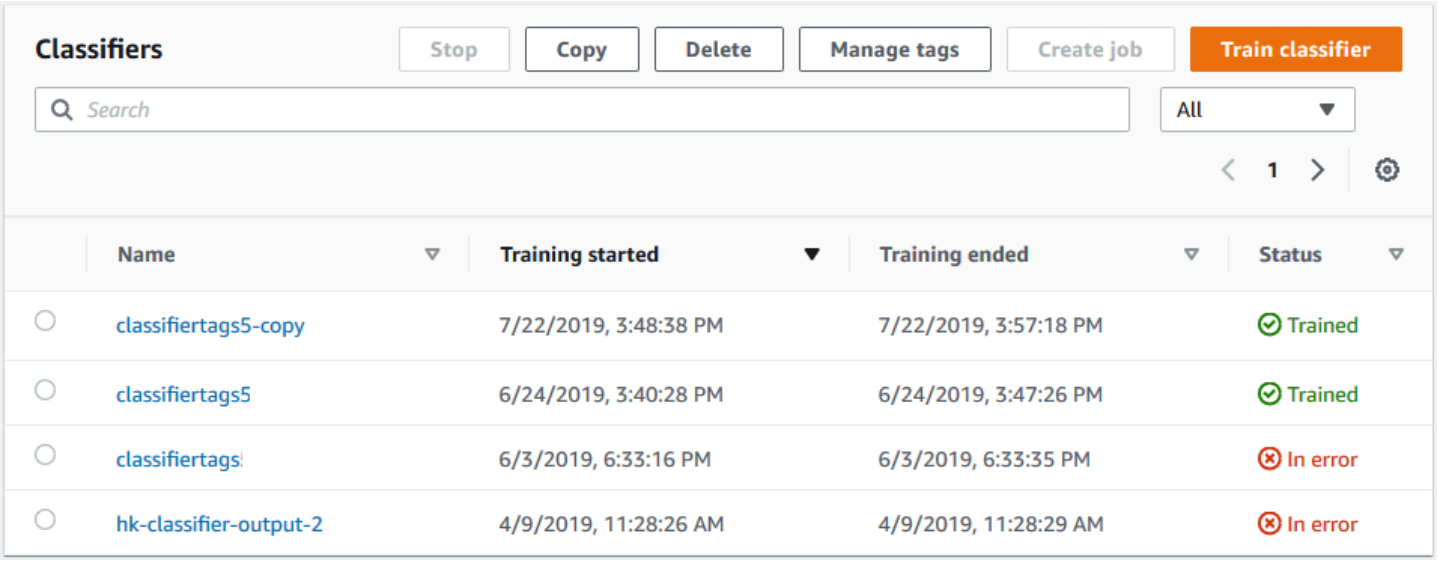
    1. Escolha a sub-rede em Sub-redes. Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.
    2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

    > **Note**
    >
    > Quando você usa uma VPC com sua tarefa de classificação, os `DataAccessRole` usados para as operações Create and Start devem ter permissões para a VPC que acessa os documentos de entrada e o bucket de saída.

17. (Opcional) Para adicionar uma tag ao classificador personalizado, insira um par de valores-chave em Tags. Escolha Adicionar Tag. Para remover esse par antes de criar o classificador, escolha Remover tag. Para obter mais informações, consulte Marcando seus Recursos .
18. Escolha Criar.

O console exibe a página Classificadores. O novo classificador aparece na tabela, mostrando `Submitted` como seu status. Quando o classificador começa a processar os documentos de treinamento, o status muda para `Training`. Quando um classificador está pronto para uso, o status muda para `Trained` ou `Trained with warnings`. Se o status for `TRAINED_WITH_WARNINGS`, revise a pasta de arquivos ignorados no Saída do treinamento do classificador.

Se o Amazon Comprehend encontrar erros durante a criação ou o treinamento, o status mudará para `In error`. Escolha uma tarefa classificadora na tabela para obter mais informações sobre o classificador, incluindo mensagens de erro.



Treine classificadores personalizados (API)

Para criar e treinar um classificador personalizado, use a CreateDocumentClassifieroperação.

Você pode monitorar o progresso da solicitação usando a DescribeDocumentClassifieroperação. Após a transição do campo `Status` para `TRAINED`, você pode usar o classificador para classificar documentos. Se o status for `TRAINED_WITH_WARNINGS`, revise a pasta de arquivos ignorados no Saída do treinamento do classificador para a operação `CreateDocumentClassifier`.

Tópicos

- Classificação personalizada de treinamento usando o AWS Command Line Interface
- Usando o AWS SDK para Java ou SDK para Python

Classificação personalizada de treinamento usando o AWS Command Line Interface

Os exemplos a seguir mostram como usar a `CreateDocumentClassifier` operação, a `DescribeDocumentClassificationJob` operação e outro classificador personalizado APIs com o. AWS CLI

O exemplo está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Crie um classificador personalizado de texto sem formatação usando a operação `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
```

Para criar um classificador personalizado nativo, forneça os seguintes parâmetros adicionais na solicitação para `create-document-classifier`.

1. DocumentType: defina o valor como SEMI_STRUCTURED_DOCUMENT.
2. Documents: a localização do S3 para os documentos de treinamento (e, opcionalmente, os documentos de teste).
3. OutputDataConfig: forneça a localização do S3 para os documentos de saída (e uma chave KMS opcional).
4. DocumentReaderConfig: campo opcional para configurações de extração de texto.

```
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
```

Obtenha informações sobre um classificador personalizado com o classificador de documentos ARN usando a operação `DescribeDocumentClassifier`.

```
aws comprehend describe-document-classifier \
     --region region \
     --document-classifier-arn arn:aws:comprehend:region:account number:document-
classifier/file name
```

Exclua um classificador personalizado usando a operação `DeleteDocumentClassifier`.

```
aws comprehend delete-document-classifier \
```

```
    --region region \
    --document-classifier-arn arn:aws:comprehend:region:account number:document-
classifier/testDelete
```

Liste todos os classificadores personalizados na conta usando a operação `ListDocumentClassifiers`.

```
aws comprehend list-document-classifiers
    --region region
```

Usando o AWS SDK para Java ou SDK para Python

Para exemplos de SDK sobre como criar e treinar um classificador personalizado, consulte Use `CreateDocumentClassifier` com um AWS SDK ou CLI.

Teste os dados de treinamento

Depois de treinar o modelo, o Amazon Comprehend testa o modelo de classificador personalizado. Se você não fornecer um conjunto de dados de teste, o Amazon Comprehend treina o modelo com 90% dos dados de treinamento. Ele reserva 10% dos dados de treinamento para usar nos testes. Se você fornecer um conjunto de dados de teste, os dados de teste devem incluir pelo menos um exemplo para cada rótulo único no conjunto de dados de treinamento.

O teste do modelo fornece métricas que podem ser usadas para estimar a precisão do modelo. O console exibe as métricas na seção Desempenho do classificador da página de Detalhes do classificador no console. Eles também são `Metrics` retornados nos campos retornados pela DescribeDocumentClassifieroperação.

No exemplo de dados de treinamento a seguir, há cinco rótulos: DOCUMENTÁRIO, DOCUMENTÁRIO, FICÇÃO_CIENTÍFICA, DOCUMENTÁRIO, COMÉDIA_ROMÂNTICA. Existem três classes únicas: DOCUMENTÁRIO, FICÇÃO_CIENTÍFICA, COMÉDIA_ROMÂNTICA.

Coluna 1	Coluna 2
DOCUMENTÁRIO	texto do documento 1
DOCUMENTÁRIO	texto do documento 2
FICÇÃO_CIENTÍFICA	texto do documento 3

Coluna 1	Coluna 2
DOCUMENTÁRIO	texto do documento 4
COMÉDIA_ROMÂNTICA	texto do documento 5

Para a divisão automática (em que o Amazon Comprehend reserva 10% dos dados de treinamento para uso em testes), se os dados de treinamento contiverem exemplos limitados de um rótulo específico, o conjunto de dados de teste poderá conter zero exemplos desse rótulo. Por exemplo, se o conjunto de dados de treinamento contiver 1.000 instâncias da classe DOCUMENTÁRIO, 900 instâncias de FICÇÃO_CIENTÍFICA e uma única instância da classe COMÉDIA_ROMÂNTICA, o conjunto de dados de teste poderá conter 100 ocorrências de DOCUMENTÁRIO e 90 de FICÇÃO_CIENTÍFICA, mas nenhuma instância de COMÉDIA_ROMÂNTICA, pois há um único exemplo disponível.

Depois de terminar de treinar seu modelo, as métricas de treinamento fornecem informações que podem ser usadas para decidir se o modelo é preciso o suficiente para suas necessidades.

Saída do treinamento do classificador

Depois que o Amazon Comprehend conclui o treinamento do modelo de classificador personalizado, ele cria arquivos de saída no local de saída do Amazon S3 que você especificou na solicitação de API ou CreateDocumentClassifierna solicitação de console equivalente.

O Amazon Comprehend cria uma matriz de confusão quando você treina um modelo de texto sem formatação ou um modelo de documento nativo. Ele pode criar arquivos de saída adicionais quando você treina um modelo de documento nativo.

Tópicos

- Matriz de confusão
- Saídas adicionais para modelos de documentos nativos

Matriz de confusão

Quando você treina um modelo de classificador personalizado, o Amazon Comprehend cria uma matriz de confusão que fornece métricas sobre o desempenho do modelo no treinamento. Essa matriz mostra uma matriz de rótulos que o modelo previu, em comparação com os rótulos reais do

documento. O Amazon Comprehend usa uma parte dos dados de treinamento para criar a matriz de confusão.

Uma matriz de confusão fornece uma indicação de quais classes poderiam usar mais dados para melhorar o desempenho do modelo. Uma classe com uma alta fração de previsões corretas tem o maior número de resultados ao longo da diagonal da matriz. Se o número na diagonal for menor, a classe terá uma fração menor de previsões corretas. Você pode adicionar mais exemplos de treinamento para essa classe e treinar o modelo novamente. Por exemplo, se 40% das amostras do rótulo A forem classificadas como rótulo D, adicionar mais amostras para o rótulo A e o rótulo D aumentará o desempenho do classificador.

Depois que o Amazon Comprehend cria o modelo classificador, a matriz de confusão fica disponível no arquivo `confusion_matrix.json` no local de saída do S3.

O formato da matriz de confusão varia, dependendo se você treinou seu classificador usando o modo multiclasse ou o multirrótulo.

Tópicos

- Matriz de confusão para o modo multiclasse
- Matriz de confusão para o modo multirrótulo

Matriz de confusão para o modo multiclasse

No modo multiclasse, as classes individuais são mutuamente exclusivas, então a classificação atribui um rótulo a cada documento. Por exemplo, um animal pode ser um cachorro ou um gato, mas não os dois ao mesmo tempo.

Considere o exemplo a seguir de uma matriz de confusão para um classificador treinado em várias 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)
```

Nesse caso, o modelo previu o seguinte:

- Um rótulo “A” foi previsto com precisão, dois rótulos “A” foram previstos incorretamente como rótulos “B” e quatro rótulos “A” foram previstos incorretamente como rótulos “Y”.
- Três rótulos “B” foram previstos com precisão e um rótulo “B” foi previsto incorretamente como um rótulo “Y”.
- Um “X” foi previsto com precisão.
- Um rótulo “Y” foi previsto com precisão, um foi previsto incorretamente como rótulo “A”, um foi previsto incorretamente como rótulo “B” e um foi previsto incorretamente como rótulo “X”.

A linha diagonal na matriz (A:A, B:B, X:X e Y:Y) mostra as previsões precisas. Os erros de predição são os valores fora da diagonal. Nesse caso, a matriz mostra as seguintes taxas de erro de predição:

- Rótulos A: 86%
- Rótulos B: 25%
- Rótulos X: 0%
- Rótulos Y: 75%

O classificador retorna a matriz de confusão como um arquivo no formato JSON. O arquivo JSON a seguir representa a matriz do exemplo anterior.

```
{
 "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"]
}
```

Matriz de confusão para o modo multirrótulo

No modo multirrótulo, a classificação pode atribuir uma ou mais classes a um documento. Considere o exemplo a seguir de uma matriz de confusão para um classificador treinado em várias classes.

Neste exemplo, há três rótulos possíveis: `Comedy`, `Action` e `Drama`. A matriz de confusão de vários rótulos cria uma matriz 2x2 para cada rótulo.

```
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)--------|
```

Nesse caso, o modelo retornou o seguinte para o rótulo de `Comedy`:

- Dois casos em que foi prevista a presença de um rótulo de `Comedy` com precisão. True positive (TP — verdadeiro positivo).
- Duas instâncias em que se previu com precisão a ausência de um rótulo de `Comedy`. True negative (TN — falso verdadeiro)
- Zero instâncias em que a presença de um rótulo de `Comedy` foi prevista incorretamente. False positive (FP — falso positivo)
- Uma instância em que foi prevista incorretamente a ausência de um rótulo de `Comedy`. False negative (FN — falso negativo)

Assim como em uma matriz de confusão multiclasse, a linha diagonal em cada matriz mostra as previsões precisas.

Nesse caso, o modelo previu rótulos de `Comedy` com precisão em 80% das vezes (TP mais TN) e os previu incorretamente em 20% das vezes (FP mais FN).

O classificador retorna a matriz de confusão como um arquivo no formato JSON. O arquivo JSON a seguir representa a matriz do exemplo anterior.

```
{
"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"]
}
```

Saídas adicionais para modelos de documentos nativos

O Amazon Comprehend pode criar arquivos de saída adicionais quando você treina um modelo de documento nativo.

Saída do Amazon Textract

Se o Amazon Comprehend invocou o Amazon Textract APIs para extrair texto para qualquer um dos documentos de treinamento, ele salvará os arquivos de saída do Amazon Textract no local de saída do S3. Ele usa a seguinte estrutura de diretórios:

- Documentos de treinamento:

  `amazon-textract-output/train/<file_name>/<page_num>/textract_output.json`
- Documentos de teste:

  `amazon-textract-output/test/<file_name>/<page_num>/textract_output.json`

O Amazon Comprehend preenche a pasta de teste se você tiver fornecido documentos de teste na solicitação da API.

Falhas de anotação de documento

O Amazon Comprehend cria os seguintes arquivos no local de saída do Amazon S3 (na pasta documentos_ignorados/) se houver alguma falha na anotação:

- failed_annotations_train.jsonl

  O arquivo existe se alguma anotação falhar nos dados de treinamento.
- failed_annotations_test.jsonl

  O arquivo existe se a solicitação incluiu dados de teste e qualquer anotação falhou nos dados de teste.

Os arquivos de anotação com falha são arquivos JSONL com o seguinte formato:

```
{
     "File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
    {"File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."
  }
```

Métricas de classificador personalizados

O Amazon Comprehend fornece métricas para ajudar você a estimar o desempenho de um classificador personalizado. O Amazon Comprehend calcula as métricas usando os dados de teste da tarefa de treinamento do classificador. As métricas representam com precisão o desempenho do modelo durante o treinamento, portanto, elas se aproximam do desempenho do modelo para classificação de dados semelhantes.

Use operações de API, como DescribeDocumentClassifierrecuperar as métricas de um classificador personalizado.

> **Note**
>
> Consulte Métricas: Precisão, recall e FScore para uma compreensão das métricas subjacentes de Precisão, Recall e pontuação F1. Essas métricas são definidas em nível de classe. O Amazon Comprehend usa a média macro para combinar essas métricas no conjunto de testes P, R e F1, conforme discutido a seguir.

Tópicos

- Métricas
- Melhorar o desempenho do seu classificador personalizado

Métricas

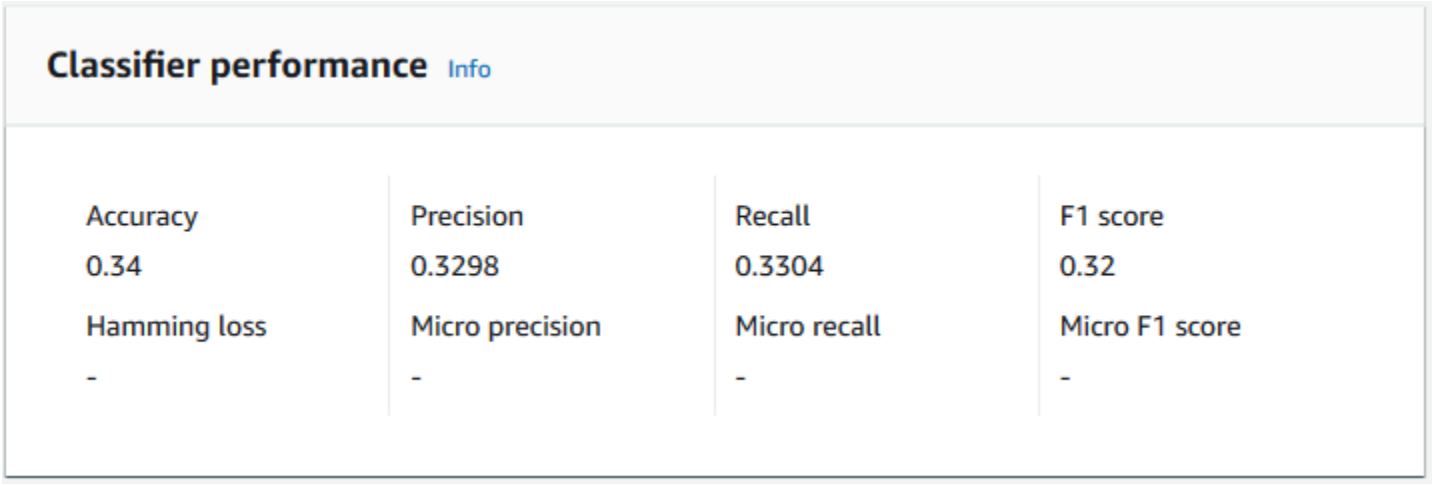
O Amazon Comprehend é compatível com as seguintes métricas:

Tópicos

- Precisão
- Precisão (precisão macro)
- Recall (recall macro)

- Pontuação F1 (macro pontuação F1)
- Perda de Hamming
- Micro precisão
- Micro recall
- Pontuação Micro F1

Para visualizar as métricas de um classificador, abra a página Detalhes do classificador no console.



Precisão

A precisão indica a porcentagem de rótulos dos dados de teste que o modelo previu com precisão. Para calcular a precisão, divida o número de rótulos previstos com precisão nos documentos de teste pelo número total de rótulos nos documentos de teste.

Por exemplo

Rótulo real	Rótulo previsto	Preciso/Incorreto
1	1	Preciso
0	1	Incorreto
2	3	Incorreto
3	3	Preciso
2	2	Preciso

Rótulo real	Rótulo previsto	Preciso/Incorreto
1	1	Preciso
3	3	Preciso

A precisão consiste no número de previsões precisas dividido pelo número geral de amostras de teste = 5/7 = 0,714 ou 71,4%

Precisão (precisão macro)

A precisão é uma medida da utilidade dos resultados do classificador nos dados do teste. É definido como o número de documentos classificados com precisão, dividido pelo número total de classificações da turma. Alta precisão significa que o classificador retornou resultados significativamente mais relevantes do que resultados irrelevantes.

A métrica de `Precision` também é conhecida como Precisão macro.

O exemplo a seguir mostra resultados de precisão para um conjunto de testes.

Rótulo	Tamanho da amostra	Precisão do rótulo
Rótulo_1	400	0.75
Rótulo_2	300	0,80
Rótulo_3	30000	0.90
Rótulo_4	20	0.50
Rótulo_5	10	0,40

A métrica de precisão (Precisão macro) para o modelo é, portanto:

```
Macro Precision = (0.75 + 0.80 + 0.90 + 0.50 + 0.40)/5 = 0.67
```

Recall (recall macro)

Isso indica a porcentagem de categorias corretas em seu texto que o modelo pode prever. Essa métrica vem da média das pontuações de recall de todos os rótulos disponíveis. O recall é uma medida do quanto os resultados do classificador estão completos para os dados do teste.

Um alto recall significa que o classificador retornou a maioria dos resultados relevantes.

A métrica de `Recall` também é conhecida como Recall macro.

O exemplo a seguir mostra resultados de recall para um conjunto de testes.

Rótulo	Tamanho da amostra	Recall de rótulo
Rótulo_1	400	0,70
Rótulo_2	300	0,70
Rótulo_3	30000	0,98
Rótulo_4	20	0,80
Rótulo_5	10	0.10

A métrica recall (Recall macro) para o modelo é, portanto:

```
Macro Recall = (0.70 + 0.70 + 0.98 + 0.80 + 0.10)/5 = 0.656
```

Pontuação F1 (macro pontuação F1)

A pontuação F1 é derivada dos valores de `Precision` e de `Recall`. Ele mede a precisão geral do classificador. A pontuação mais alta é 1 e a mais baixa é 0.

O Amazon Comprehend calcula a Macro pontuação F1. É a média não ponderada das pontuações F1 dos rótulos. Utilizar o seguinte conjunto de testes como exemplo:

Rótulo	Tamanho da amostra	Pontuação F1 do rótulo
Rótulo_1	400	0,724

Rótulo	Tamanho da amostra	Pontuação F1 do rótulo
Rótulo_2	300	0,824
Rótulo_3	30000	0,94
Rótulo_4	20	0,62
Rótulo_5	10	0,16

A Macro F1 Score (macro pontuação F1) para o modelo é calculada da seguinte forma:

```
Macro F1 Score = (0.724 + 0.824 + 0.94 + 0.62 + 0.16)/5 = 0.6536
```

Perda de Hamming

A fração de rótulos previstos incorretamente. Também é visto como a fração de rótulos incorretos em comparação com o número total de rótulos. Pontuações próximas a zero são melhores.

Micro precisão

Original:

Semelhante à métrica de precisão, exceto que a micro precisão é baseada na pontuação geral de todas as pontuações de precisão somadas.

Micro recall

Semelhante à métrica de recall, exceto que o micro recall é baseado na pontuação geral de todas as pontuações de recall somadas.

Pontuação Micro F1

A Pontuação Micro F1 é uma combinação das métricas de micro precisão e micro recall.

Melhorar o desempenho do seu classificador personalizado

As métricas fornecem uma visão sobre o desempenho do seu classificador personalizado durante uma tarefa de classificação. Se as métricas forem baixas, o modelo de classificação pode não ser eficaz para seu caso de uso. Você tem várias opções para melhorar o desempenho do classificador:

1. Em seus dados de treinamento, dê exemplos concretos que definam uma separação clara das categorias. Por exemplo, forneça documentos que usem exclusivos words/sentences para representar a categoria.
2. Adicione mais dados para rótulos pouco representados em seus dados de treinamento.
3. Tente reduzir a distorção nas categorias. Se o maior rótulo em seus dados tiver mais de 10 vezes os documentos no menor rótulo, tente aumentar o número de documentos para o menor rótulo. Reduza a taxa de inclinação para no máximo 10:1 entre classes altamente representadas e menos representadas. Você também pode remover documentos de entrada das classes altamente representadas.

Executar análises em tempo real

Depois de treinar um classificador personalizado, você pode classificar documentos usando análise em tempo real. A análise em tempo real usa um único documento como entrada e retorna os resultados de forma síncrona. A classificação personalizada aceita uma variedade de tipos de documentos como entradas para análise em tempo real. Para obter detalhes, consulte Entradas para análise personalizada em tempo real.

Se você planeja analisar arquivos de imagem ou documentos PDF digitalizados, sua política do IAM deve conceder permissões para usar dois métodos da API Amazon Textract DetectDocumentText (AnalyzeDocumente). O Amazon Comprehend invoca esses métodos durante a extração de texto. Para visualizar um exemplo de política, consulte Permissões necessárias para executar ações de análise de documentos.

Você deve criar um endpoint para executar análises em tempo real usando um modelo de classificação personalizado.

Tópicos

- Análise em tempo real para classificação personalizada (console)
- Análise em tempo real para classificação personalizada (API)
- Saídas para análise em tempo real

Análise em tempo real para classificação personalizada (console)

Use o console do Amazon Comprehend para executar análises em tempo real usando um modelo de classificação personalizado.

Você cria um endpoint para executar a análise em tempo real. Um endpoint inclui recursos gerenciados que disponibilizam seu modelo personalizado para inferência em tempo real.

Para informações sobre o fornecimento do throughput do endpoint e os custos associados, consulte Utilizar endpoints do Amazon Comprehend.

Tópicos

- Criar um endpoint para classificação personalizada
- Executar classificação personalizada em tempo real

Criar um endpoint para classificação personalizada

Criar um endpoint (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Endpoints e, depois, o botão Criar endpoint. A tela Criar endpoint é aberta.
3. Dê um nome ao endpoint. O nome deve ser exclusivo na sua conta e região atual.
4. Escolha um modelo personalizado ao qual você deseja anexar o novo endpoint. No menu suspenso, pesquise pelo nome do modelo.

   > **Note**
   >
   > Crie um modelo antes de poder anexar um endpoint a ele. Se ainda não tiver um modelo, consulte Modelos de classificação de treinamento.

5. (Opcional) para adicionar uma tag ao endpoint, insira um par de valores-chave em Tags e escolha Adicionar tag. Para remover esse par antes de criar o endpoint, escolha Remover tag
6. Insira o número de unidades de inferência (IUs) a serem atribuídas ao ponto final. Cada IU representa um throughput de 100 caracteres por segundo para até dois documentos por segundo. Para mais informações sobre o throughput do endpoint, consulte Utilizar endpoints do Amazon Comprehend.
7. (Opcional) Se você estiver criando um novo endpoint, terá a opção de usar o estimador de IU. Dependendo do throughput ou do número de caracteres que deseja analisar por segundo, pode ser difícil saber quantas IUs você precisa. Essa etapa opcional pode ajudá-lo a determinar o número de solicitações IUs a serem solicitadas.

8. Em Resumo da compra, analise seu custo estimado de endpoint por hora, dia e mês.
9. Marque a caixa de seleção se você está ciente de que sua conta incorre em cobranças pelo endpoint desde o momento em que ele é iniciado até que você o exclua.
10. Escolha Criar endpoint

Executar classificação personalizada em tempo real

Após criar um endpoint, você poderá executar análises em tempo real usando seu modelo personalizado. Há duas maneiras de executar análises em tempo real a partir do console. Você pode inserir texto ou fazer upload de um arquivo, conforme mostrado a seguir.

Executar análises em tempo real usando um modelo personalizado (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Análise em tempo real.
3. Em Tipo de entrada, escolha Personalizado para o Tipo de análise.
4. Em Tipo de modelo personalizado, escolha Classificação personalizada.
5. Em Endpoint, escolha o endpoint que deseja usar. Esse endpoint está vinculado a um modelo personalizado específico.
6. Para especificar os dados de entrada para análise, insira um texto ou carregue um arquivo.
   - Inserir texto:
     a. Escolha Texto de entrada.
     b. Insira o texto que deseja analisar.
   - Como carregar um arquivo:
     a. Escolha Fazer upload de arquivo e insira o nome do arquivo para fazer o upload.
     b. (Opcional) Em Ações de leitura avançadas, você pode substituir as ações padrão para extração de texto. Para obter detalhes, consulte Configurar opções de extração de texto

   Para melhores resultados, combine o tipo de entrada com o tipo de modelo do classificador. O console exibirá um aviso se você enviar um documento nativo para um modelo de texto sem formatação ou texto sem formatação para um modelo de documento nativo. Para obter mais informações, consulte Modelos de classificação de treinamento.

7. Escolha Analisar. O Amazon Comprehend analisa os dados de entrada usando seu modelo personalizado. O Amazon Comprehend exibe as classes descobertas, junto com uma avaliação de confiança para cada classe.

Análise em tempo real para classificação personalizada (API)

Você pode usar a API do Amazon Comprehend para executar a classificação em tempo real com um modelo personalizado. Primeiro, crie um endpoint para executar a análise em tempo real. Depois de criá-lo, execute a classificação em tempo real.

Os exemplos nesta seção usam formatos de comando para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Para informações sobre o fornecimento do throughput do endpoint e os custos associados, consulte Utilizar endpoints do Amazon Comprehend.

Tópicos

- Criar um endpoint para classificação personalizada
- Executar classificação personalizada em tempo real

Criar um endpoint para classificação personalizada

O exemplo a seguir mostra a operação da CreateEndpointAPI usando AWS CLI o.

```
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
```

O Amazon Comprehend responde com o seguinte:

```
{
   "EndpointArn": "Arn"
}
```

Executar classificação personalizada em tempo real

Depois de criar um endpoint para seu modelo de classificação personalizado, você usa o endpoint para executar a operação da ClassifyDocumentAPI. Você pode fornecer entrada de texto usando os parâmetros`text` ou `bytes`. Insira os outros tipos de entrada usando o parâmetro `bytes`.

Para arquivos de imagem e arquivos PDF, você pode usar o parâmetro `DocumentReaderConfig` para substituir as ações padrão de extração de texto. Para obter detalhes, consulte Configurar opções de extração de texto

Para melhores resultados, combine o tipo de entrada com o tipo de modelo do classificador. A resposta da API inclui um aviso se você enviar um documento nativo para um modelo de texto sem formatação ou um arquivo de texto sem formatação para um modelo de documento nativo. Para obter mais informações, consulte Modelos de classificação de treinamento.

Usando o AWS Command Line Interface

Os exemplos a seguir demonstram como usar o comando CLI classify-document.

Classifique o texto usando o AWS CLI

O exemplo a seguir executa a classificação em tempo real em um bloco de texto.

```
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.'
```

O Amazon Comprehend responde com o seguinte:

```
{
    "Classes": [
       {
          "Name": "string",
          "Score": 0.9793661236763
```

```
        }
    ]
}
```

Classifique um documento semiestruturado usando o AWS CLI

Para analisar a classificação personalizada de um arquivo PDF, Word ou de imagem, execute o comando `classify-document` com o arquivo de entrada no parâmetro `bytes`.

O exemplo a seguir usa uma imagem como arquivo de entrada. Ele usa a opção `fileb` para codificar em base 64 os bytes do arquivo de imagem. Para obter mais informações, consulte Objetos binários grandes no Guia AWS Command Line Interface do usuário.

Esse exemplo também passa um arquivo JSON chamado `config.json` para definir as opções de extração de texto.

```
$ aws comprehend classify-document \
> --endpoint-arn arn \
> --language-code en \
> --bytes fileb://image1.jpg   \
> --document-reader-config file://config.json
```

O arquivo config.json contém o conteúdo a seguir.

```
{
   "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
   "DocumentReadAction": "TEXTRACT_DETECT_DOCUMENT_TEXT"
}
```

O Amazon Comprehend responde com o seguinte:

```
{
    "Classes": [
       {
          "Name": "string",
          "Score": 0.9793661236763
       }
    ]
```

```
}
```

Para obter mais informações, consulte a ClassifyDocumentReferência da API Amazon Comprehend.

Saídas para análise em tempo real

Saídas para entradas de texto

Para entradas de texto, a saída inclui a lista de classes ou rótulos identificados pela análise do classificador. O exemplo a seguir mostra uma lista com duas classes.

```
"Classes": [
  {
     "Name": "abc",
     "Score": 0.2757999897003174,
     "Page": 1
  },
  {
    "Name": "xyz",
    "Score": 0.2721000015735626,
    "Page": 1
  }
]
```

Saídas para entradas semiestruturadas

Para um documento de entrada semiestruturado ou um arquivo de texto, a saída pode incluir os seguintes campos adicionais:

- DocumentMetadata — Extração de informações sobre o documento. Os metadados incluem uma lista de páginas no documento, com o número de caracteres extraídos de cada página. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- DocumentType — O tipo de documento para cada página no documento de entrada. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- Erros: erros no nível da página que o sistema detectou ao processar o documento de entrada. O campo estará vazio se o sistema não encontrar erros.
- Avisos: avisos detectados durante o processamento do documento de entrada. A resposta inclui um aviso se houver uma incompatibilidade entre o tipo de documento de entrada e o tipo de

modelo associado ao endpoint que você especificou. O campo ficará vazio se o sistema não tiver gerado nenhum aviso.

Para obter mais detalhes sobre esses campos de saída, consulte a ClassifyDocumentReferência da API Amazon Comprehend.

O exemplo a seguir mostra a saída para um documento de entrada PDF nativo de uma página.


```
{
  "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"
      }
  ]
}
```

Executar tarefas assíncronas

Depois de treinar um classificador personalizado, use tarefas assíncronas para analisar documentos grandes ou vários documentos em um lote.

A classificação personalizada aceita uma variedade de tipos de documentos de entrada. Para obter detalhes, consulte Entradas para análise personalizada assíncrona.

Se você planeja analisar arquivos de imagem ou documentos PDF digitalizados, sua política do IAM deve conceder permissões para usar dois métodos da API Amazon Textract DetectDocumentText (AnalyzeDocumente). O Amazon Comprehend invoca esses métodos durante a extração de texto. Para visualizar um exemplo de política, consulte Permissões necessárias para executar ações de análise de documentos.

Para classificação de documentos semiestruturados (arquivos de imagem, PDF ou Docx) usando um modelo de texto sem formatação, use o formato de entrada `one document per file`. Além disso, inclua o `DocumentReaderConfig` parâmetro em sua StartDocumentClassificationJobsolicitação.

Tópicos

- Formatos de arquivo para análise assíncrona
- Tarefas de análise para classificação personalizada (console)
- Tarefas de análise para classificação personalizada (API)
- Saídas para trabalhos de análise assíncrona

Formatos de arquivo para análise assíncrona

Ao executar a análise assíncrona com seu modelo, você tem uma escolha de formatos para documentos de entrada: `One document per line` ou `one document per file`. O formato usado depende do tipo de documento que deseja analisar, conforme descrito na tabela a seguir.

Descrição	Formato
A entrada contém vários arquivos. Cada arquivo contém um documento de entrada. Esse formato é ideal para coleções de documentos grandes, como artigos de jornal ou artigos científicos.	Um documento por arquivo

Descrição	Formato
Além disso, use esse formato para documentos semiestruturados (arquivos de imagem, PDF ou Docx) usando um classificador de documentos nativo.	
A entrada é um ou mais arquivos. Cada linha no arquivo é um documento de entrada separado. Esse formato é ideal para documentos curtos, como mensagens de texto ou publicações em redes sociais.	Um documento por linha

Um documento por arquivo

Com o formato `one document per file`, cada arquivo representa um documento de entrada.

Um documento por linha

Com o formato `One document per line`, cada documento é colocado em uma linha separada e nenhum cabeçalho é usado. O rótulo não está incluído em cada linha (já que você ainda não conhece o rótulo do documento). Cada linha do arquivo (o final do documento individual) deve terminar com uma alimentação de linha (LF,\n), uma devolução de carro (CR,\ r) ou ambas (CRLF, \r \n). Não use o separador de linha UTF-8 (u+2028) para finalizar uma linha.

O exemplo a seguir mostra o formato do arquivo de entrada.

```
Text of document 1 \n
Text of document 2 \n
Text of document 3 \n
Text of document 4 \n
```

Para qualquer formato, use a codificação UTF-8 para arquivos de texto. Depois de preparar os arquivos, coloque-os no bucket do S3 que está usando para os dados de entrada.

Ao iniciar uma tarefa de classificação, você especifica esse local do Amazon S3 para seus dados de entrada. O URI deve estar na mesma região que a API do endpoint que você está chamando. O URI pode apontar para um único arquivo (como ao usar o método “um documento por linha”) ou pode ser o prefixo de uma coleção de arquivos de dados.

Por exemplo, se você usar o URI `S3://bucketName/prefix`, se o prefixo for um único arquivo, o Amazon Comprehend usará esse arquivo como entrada. Se mais de um arquivo começar com o prefixo, o Amazon Comprehend usará todos eles como entrada.

Conceda ao Amazon Comprehend acesso ao bucket do S3 contendo sua coleção de documentos e arquivos de saída. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Tarefas de análise para classificação personalizada (console)

Depois de criar e treinar um classificador de documentos personalizado, use o console para executar tarefas de classificação personalizadas com o modelo.

Criar uma tarefa de classificação personalizada (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Tarefas de análise e, em seguida, Criar tarefa.
3. Dê um nome ao trabalho de classificação. O nome deve ser exclusivo na sua conta e na sua região atual.
4. Em Tipo de análise, escolha Classificação personalizada.
5. Em Selecionar classificador, escolha o classificador personalizado a ser usado.
6. (Opcional) Se você optar por criptografar os dados que o Amazon Comprehend usa ao processar sua tarefa, escolha Criptografia de tarefas. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta.

   - Se estiver usando uma chave associada à conta atual, escolha o ID da chave para o ID da chave KMS.
   - Se estiver usando uma chave associada a uma conta diferente, insira o ARN do ID da chave em ARN da chave KMS.

> **Note**
>
> Para obter mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte Serviço de gerenciamento de chaves (KMS).

7. Em Dados de entrada, insira a localização do bucket do Amazon S3 contendo seus documentos de entrada ou navegue até ele escolhendo Procurar no S3. Esse bucket deve estar na mesma região da API que você está chamando. O perfil do IAM que você está usando para obter permissões de acesso à tarefa de classificação deve ter permissões de leitura para o bucket do S3.

   Para alcançar o mais alto nível de precisão no treinamento de um modelo, combine o tipo de entrada com o tipo de modelo do classificador. A tarefa classificadora retornará um aviso se você enviar documentos nativos para um modelo de texto sem formatação ou documentos de texto sem formatação para um modelo de documento nativo. Para obter mais informações, consulte Modelos de classificação de treinamento.

8. (Opcional) Em Formato de entrada, escolha o formato dos documentos de entrada. O formato pode ser um documento por arquivo ou um documento por linha em um único arquivo. Um documento por linha se aplica somente a documentos de texto.

9. (Opcional) Para Modo de leitura do documento, você pode substituir as ações padrão de extração de texto. Para obter mais informações, consulte Configurar opções de extração de texto.

10. Em Dados de saída, insira a localização do bucket do Amazon S3 em que o Amazon Comprehend deve gravar os dados de saída da tarefa ou navegar até elas escolhendo Procurar no S3 Esse bucket deve estar na mesma região da API que você está chamando. O perfil do IAM que você está usando para obter permissões de acesso à tarefa de classificação deve ter permissões de gravação para o bucket do S3.

11. (Opcional) Se optar por criptografar o resultado de saída da sua tarefa, escolha Criptografia. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta.

    - Se você estiver usando uma chave associada à conta atual, escolha o alias ou ID da chave para o ID da chave KMS.
    - Se você estiver usando uma chave associada a uma conta diferente, insira o ARN do alias ou ID da chave em ID da chave KMS.

12. (Opcional) Para lançar seus recursos no Amazon Comprehend a partir de uma VPC, insira o ID da VPC em VPC ou escolha o ID na lista suspensa.

    1. Escolha a sub-rede em Sub-rede(s). Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.

2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

> **Note**
>
> Quando você usa uma VPC com sua tarefa de classificação, o `DataAccessRole` usado para as operações Create and Start deve conceder permissões à VPC que acessa o bucket de saída.

13. Escolha Criar tarefa para criar a tarefa de classificação de documentos.

Tarefas de análise para classificação personalizada (API)

Depois de criar e treinar um classificador de documentos personalizado, use o classificador para executar tarefas de análise.

Use a StartDocumentClassificationJoboperação para começar a classificar documentos sem rótulos. Você especifica o bucket do S3 que contém os documentos de entrada, o bucket do S3 para os documentos de saída e o classificador a ser usado.

Para alcançar o mais alto nível de precisão no treinamento de um modelo, combine o tipo de entrada com o tipo de modelo do classificador. A tarefa classificadora retornará um aviso se você enviar documentos nativos para um modelo de texto sem formatação ou documentos de texto sem formatação para um modelo de documento nativo. Para obter mais informações, consulte Modelos de classificação de treinamento.

StartDocumentClassificationJobé assíncrono. Depois de iniciar o trabalho, use a DescribeDocumentClassificationJoboperação para monitorar seu progresso. Quando o campo `Status` na resposta exibir `COMPLETED`, você poderá acessar a saída no local especificado.

Tópicos

- Usando o AWS Command Line Interface
- Usando o AWS SDK para Java ou SDK para Python

Usando o AWS Command Line Interface

Os exemplos a seguir são a `StartDocumentClassificationJob` operação e outro classificador personalizado APIs com o. AWS CLI

Os exemplos a seguir usam o formato de comando para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Execute uma tarefa de classificação personalizada usando a operação `StartDocumentClassificationJob`.

```
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
```

Obtenha informações sobre um classificador personalizado com o ID da tarefa usando a operação `DescribeDocumentClassificationJob`.

```
aws comprehend describe-document-classification-job \
     --region region \
     --job-id job id
```

Liste todas as tarefas de classificação personalizada em sua conta usando a operação `ListDocumentClassificationJobs`.

```
aws comprehend list-document-classification-jobs
     --region region
```

Usando o AWS SDK para Java ou SDK para Python

Para exemplos de SDK sobre como iniciar um classificador personalizado, consulte Use `StartDocumentClassificationJob` com um AWS SDK ou CLI.

Saídas para trabalhos de análise assíncrona

Depois que uma tarefa de análise é concluída, ela armazena os resultados no bucket do S3 que você especificou na solicitação.

Saídas para entradas de texto

Para qualquer formato de documentos de entrada de texto (com várias classes ou com vários rótulos), a saída da tarefa consiste em um único arquivo chamado `output.tar.gz`. É um arquivo compactado contendo um arquivo de texto com a saída.

Saída multiclasse

Quando você usa um classificador treinado no modo multiclasse, seus resultados exibem as `classes`. Cada uma dessas `classes` é a classe usada para criar o conjunto de categorias ao treinar seu classificador.

Para obter mais detalhes sobre esses campos de saída, consulte a ClassifyDocumentReferência da API Amazon Comprehend.

Os exemplos a seguir usam as seguintes classes mutuamente exclusivas.

```
DOCUMENTARY
SCIENCE_FICTION
ROMANTIC_COMEDY
SERIOUS_DRAMA
OTHER
```

Se o formato dos dados de entrada for um documento por linha, o arquivo de saída conterá uma linha para cada linha na entrada. Cada linha inclui o nome do arquivo, o número da linha com base em zero da linha de entrada e a classe ou classes encontradas no documento. Termina com a confiança que o Amazon Comprehend tem de que a instância individual foi classificada corretamente.

Por exemplo:

```
{"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}]}
```

Se o formato dos dados de entrada for um documento por arquivo, o arquivo de saída conterá uma linha para cada documento. Cada linha tem o nome do arquivo e a classe ou classes encontradas no documento. Termina com a confiança de que o Amazon Comprehend classificou a instância individual com precisão.

Por exemplo:

```
{"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}]}
```

Saída multirrótulo

Quando você usa um classificador treinado no modo multirrótulo, seus resultados exibem os `labels`. Cada um desses `labels` é o rótulo usado para criar o conjunto de categorias ao treinar seu classificador.

Os exemplos a seguir usam esses rótulos exclusivos.

```
SCIENCE_FICTION
ACTION
DRAMA
COMEDY
ROMANCE
```

Se o formato dos dados de entrada for um documento por linha, o arquivo de saída conterá uma linha para cada linha na entrada. Cada linha inclui o nome do arquivo, o número da linha com base em zero da linha de entrada e a classe ou classes encontradas no documento. Termina com a confiança que o Amazon Comprehend tem de que a instância individual foi classificada corretamente.

Por exemplo:

```
{"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}]}
```

Se o formato dos dados de entrada for um documento por arquivo, o arquivo de saída conterá uma linha para cada documento. Cada linha tem o nome do arquivo e a classe ou classes encontradas no documento. Termina com a confiança de que o Amazon Comprehend classificou a instância individual com precisão.

Por exemplo:

```
{"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}]}
```

Saídas para documentos de entrada semiestruturados

Para documentos de entrada semiestruturados, a saída pode incluir os seguintes campos adicionais:

- DocumentMetadata — Extração de informações sobre o documento. Os metadados incluem uma lista de páginas no documento, com o número de caracteres extraídos de cada página. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- DocumentType — O tipo de documento para cada página no documento de entrada. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- Erros: erros no nível da página que o sistema detectou ao processar o documento de entrada. O campo estará vazio se o sistema não encontrar erros.

Para obter mais detalhes sobre esses campos de saída, consulte a ClassifyDocumentReferência da API Amazon Comprehend.

O exemplo a seguir mostra a saída de um arquivo PDF digitalizado de duas páginas.


```
[{ #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"
```

```
}]
```

Reconhecimento de entidade personalizado

O reconhecimento personalizado de entidades amplia a capacidade do Amazon Comprehend, ajudando você a identificar seus novos tipos de entidade específicos que não estão nos tipos de entidade genéricos predefinidos. Isso significa que você pode analisar documentos e extrair entidades, como códigos de produtos ou entidades específicas de negócios, que atendam às suas necessidades específicas.

Criar um reconhecedor personalizado de entidades preciso por conta própria pode ser um processo complexo, exigindo a preparação de grandes conjuntos de documentos de treinamento anotados manualmente e a seleção dos algoritmos e parâmetros certos para o treinamento de modelos. O Amazon Comprehend ajuda a reduzir a complexidade fornecendo anotações automáticas e desenvolvimento de modelos para criar um modelo personalizado de reconhecimento de entidades.

Criar um modelo personalizado de reconhecimento de entidades é uma abordagem mais eficaz do que usar correspondências de strings ou expressões regulares para extrair entidades de documentos. Por exemplo, para extrair nomes de ENGENHEIROS em um documento, é difícil enumerar todos os nomes possíveis. Além disso, sem contexto, é difícil distinguir entre nomes de ENGENHEIROS e nomes de ANALISTAS. Um modelo personalizado de reconhecimento de entidades pode aprender o contexto em que esses nomes provavelmente aparecerão. Além disso, a correspondência de strings não detectará entidades que tenham erros de digitação ou sigam novas convenções de nomenclatura, embora isso seja possível usando um modelo personalizado.

Você tem duas opções para criar um modelo personalizado:

1. Anotações: forneça um conjunto de dados contendo entidades anotadas para treinamento de modelos.
2. Listas de entidades (em texto simples apenas): forneça uma lista de entidades e seu rótulo de tipo (como `PRODUCT_CODES` e um conjunto de documentos não anotados contendo essas entidades) para treinamento de modelos.

Ao criar um reconhecedor personalizado de entidades usando arquivos PDF anotados, você pode usar esse reconhecedor com vários formatos de arquivo de entrada: texto simples, arquivos de imagem (JPG, PNG, TIFF), arquivos PDF e documentos do Word, sem a necessidade de pré-processamento ou nivelamento de documentos. O Amazon Comprehend não oferece suporte à anotação de arquivos de imagem ou documentos do Word.

> **Note**
>
> Um reconhecedor personalizado de entidades que use arquivos PDF anotados oferece suporte somente a documentos em inglês.

Você pode treinar um modelo em até 25 entidades personalizadas ao mesmo tempo. Para obter mais detalhes, consulte a página Diretrizes e cotas.

Depois que seu modelo for treinado, você poderá usá-lo para detecção de entidades em tempo real e em trabalhos de detecção de entidades.

Tópicos

- Preparando dados de treinamento do reconhecedor de entidades
- Treinamento de modelos reconhecedores de entidades personalizados
- Executar análise de reconhecedor personalizado em tempo real
- Executar trabalhos de análise para reconhecimento personalizado de entidades

Preparando dados de treinamento do reconhecedor de entidades

Para treinar um modelo de reconhecimento personalizado de entidade bem-sucedido, é importante fornecer ao treinador de modelos dados de alta qualidade como entrada. Sem bons dados, o modelo não aprenderá a identificar entidades corretamente.

Você pode escolher uma das duas formas de fornecer dados ao Amazon Comprehend para treinar um modelo personalizado de reconhecimento de entidades:

- Lista de entidades: lista as entidades específicas para que o Amazon Comprehend possa treinar para identificar suas entidades personalizadas. Observação: as listas de entidades só podem ser usadas para documentos de texto simples.
- Anotações: fornece a localização de suas entidades em vários documentos para que o Amazon Comprehend possa treinar tanto na entidade quanto em seu contexto. Para criar um modelo para analisar arquivos de imagem ou documentos do Word, você deve treinar seu reconhecedor usando anotações em PDF. PDFs

Em ambos os casos, o Amazon Comprehend aprende sobre o tipo de documentos e o contexto em que as entidades ocorrem e cria um reconhecedor que pode ser generalizado para detectar as novas entidades quando você analisa documentos.

Ao criar um modelo personalizado (ou treinar uma nova versão), você pode fornecer um conjunto de dados de teste. Se você não fornecer dados de teste, o Amazon Comprehend reserva 10% dos documentos de entrada para testar o modelo. O Amazon Comprehend treina o modelo com os documentos restantes.

Se você fornecer um conjunto de dados de teste para o seu conjunto de treinamento de anotações, os dados de teste devem incluir pelo menos uma anotação para cada um dos tipos de entidade especificados na solicitação de criação.

Tópicos

- Quando usar anotações em vez de listas de entidades
- Listas de entidades (somente texto simples)
- Anotações

Quando usar anotações em vez de listas de entidades

Criar anotações dá mais trabalho do que criar uma lista de entidades, mas o modelo resultante pode ser significativamente mais preciso. Usar uma lista de entidades é mais rápido e menos trabalhoso, mas os resultados são menos refinados e menos precisos. Isso ocorre porque as anotações fornecem mais contexto para o Amazon Comprehend usar ao treinar o modelo. Sem esse contexto, o Amazon Comprehend terá um número maior de falsos positivos ao tentar identificar as entidades.

Há cenários em que faz mais sentido para os negócios evitar grandes despesas e workload usando anotações. Por exemplo, o nome John Johnson é importante para sua pesquisa, mas, se essa é a pessoa exata, não é relevante. Ou as métricas, ao usar a lista de entidades, são boas o suficiente para fornecer os resultados do reconhecedor de que você precisa. Nesses casos, usar uma lista de entidades em vez disso pode ser a opção mais eficaz.

Recomendamos o uso do modo de anotações nos seguintes casos:

- Se você planeja executar inferências para arquivos de imagem ou documentos do Word. PDFs Nesse cenário, você treina um modelo usando arquivos PDF anotados e usa o modelo para executar trabalhos de inferência para arquivos de imagem e documentos do PDFs Word.

- Quando o significado das entidades pode ser ambíguo e dependente do contexto. Por exemplo, o termo Amazon pode se referir ao rio no Brasil ou ao varejista online Amazon.com. Ao criar um reconhecedor personalizado de entidades para identificar entidades comerciais como a Amazon, você deve usar anotações em vez de uma lista de entidades, pois esse método é mais capaz de usar o contexto para encontrar entidades.
- Quando você se sentir confortável em configurar um processo para adquirir anotações, o que pode exigir algum esforço.

Recomendamos usar uma lista de entidades nos seguintes casos:

- Quando você já tem uma lista de entidades ou quando é relativamente fácil compor uma lista abrangente de entidades. Se você usa uma lista de entidades, a lista deve estar completa ou, pelo menos, abranger a maioria das entidades válidas que podem aparecer nos documentos que você fornece para treinamento.
- Geralmente, para usuários iniciantes, é recomendável usar uma lista de entidades, porque isso exige menos esforço do que criar anotações. No entanto, é importante observar que o modelo treinado pode não ser tão preciso quanto se você usasse anotações.

Listas de entidades (somente texto simples)

Para treinar um modelo usando uma lista de entidades, você fornece duas informações: uma lista dos nomes das entidades com seus tipos personalizados de entidade correspondentes e uma coleção de documentos sem anotações nos quais você espera que suas entidades apareçam.

Quando você fornece uma lista de entidades, o Amazon Comprehend usa um algoritmo inteligente para detectar ocorrências da entidade nos documentos e servir como base para o treinamento do modelo de reconhecimento personalizado de entidades.

Para listas de entidades, forneça pelo menos 25 correspondências de entidades por tipo de entidade na lista de entidades.

Uma lista de entidades para reconhecimento personalizado de entidades precisa de um arquivo de valores separados por vírgula (CSV) com as seguintes colunas:

- Texto: o texto de um exemplo de entrada exatamente como visto no corpo do documento anexo.

- Tipo: o tipo de entidade definido pelo cliente. Os tipos de entidade devem ter uma sequência de caracteres em letras maiúsculas separadas por sublinhado, como GERENTE ou GERENTE_SENIOR. É possível treinar até 25 tipos de entidades por modelo.

O arquivo `documents.txt` contém quatro linhas:

```
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.
```

O arquivo CSV com a lista de entidades tem as seguintes linhas:

```
Text, Type
Jo Brown, ENGINEER
John Doe, ENGINEER
Jane Smith, MANAGER
```

> **Note**
>
> Na lista de entidades, a entrada de Emilio Johnson não está presente porque não contém a entidade ENGENHEIRO ou GERENTE.

Como criar seus arquivos de dados

É importante que sua lista de entidades esteja em um arquivo CSV corretamente configurado para minimizar a chance de haver problemas com seu arquivo de lista de entidades. Para configurar seu arquivo CSV manualmente, o seguinte deve ser verdadeiro:

- A codificação UTF-8 deve ser especificada explicitamente, mesmo que seja usada como padrão na maioria dos casos.
- Ele deve incluir os nomes de colunas: `Type` e `Text`.

É altamente recomendável que os arquivos de entrada CSV sejam gerados programaticamente para evitar possíveis problemas.

O exemplo a seguir usa Python para gerar um CSV para as anotações apresentadas acima:

```
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"])
```

Práticas recomendadas

Há vários aspectos a levar em consideração para obter o melhor resultado ao usar uma lista de entidades, inclusive:

- A ordem das entidades na sua lista não afeta o treinamento do modelo.
- Use itens da lista de entidades que abranjam de 80% a 100% dos exemplos de entidades positivas mencionados no corpo de documentos não anotados.
- Remova palavras e frases comuns para evitar exemplos de entidades que correspondam a entidades que não sejam entidades no corpo do documento. Até mesmo algumas correspondências incorretas podem afetar significativamente a precisão do modelo resultante. Por exemplo, uma palavra como the na lista de entidades resultará em um grande número de correspondências que provavelmente não são as entidades que você está procurando e, portanto, afetarão significativamente sua precisão.
- Os dados de entrada não devem conter duplicatas. A presença de amostras duplicadas pode resultar na contaminação do conjunto de testes e, consequentemente, afetar negativamente o processo de treinamento, as métricas e o comportamento do modelo.
- Forneça documentos que tenham a maior semelhança possível com os casos de uso reais. Não use dados fictícios ou dados sintetizados para sistemas de produção. Os dados de entrada devem ser tão diversos quanto possível para evitar sobreajustes e ajudar o modelo subjacente a fazer uma generalização melhor em exemplos reais.
- A lista de entidades diferencia maiúsculas de minúsculas e não há suporte para expressões regulares. No entanto, o modelo treinado geralmente é capaz de reconhecer entidades, mesmo que elas não correspondam exatamente ao caso fornecido na lista de entidades.
- Se você tiver uma entidade que seja uma substring de outra entidade (como “Smith” e “Jane Smith”), forneça ambas na lista de entidades.

Veja mais sugestões em Melhorar o desempenho do reconhecedor de entidade personalizado

Anotações

As anotações rotulam entidades no contexto, associando os tipos personalizados de entidade aos locais nos quais elas ocorrem nos seus documentos de treinamento.

Ao enviar anotações junto com seus documentos, você pode aumentar a precisão do modelo. Com as Anotações, você não está simplesmente fornecendo a localização da entidade que está procurando, mas também fornecendo um contexto mais preciso para a entidade personalizada que está procurando.

Por exemplo, se você estiver procurando pelo nome John Johnson, com o tipo de entidade JUIZ, fornecer a sua anotação pode ajudar o modelo a descobrir que a pessoa que você deseja encontrar é um juiz. Se conseguir usar o contexto, o Amazon Comprehend não encontrará pessoas chamadas John Johnson que sejam advogados ou testemunhas. Sem as anotações, o Amazon Comprehend criará sua própria versão de uma anotação, mas não será tão eficaz em incluir apenas juízes. O fornecimento de suas próprias anotações pode ajudar a obter melhores resultados e gerar modelos capazes de aproveitar melhor o contexto ao extrair entidades personalizadas.

Tópicos

- Número mínimo de anotações
- Práticas recomendadas de anotação
- Arquivos de anotação em texto simples
- Arquivos de anotação em PDF
- Anotação de arquivos PDF

Número mínimo de anotações

O número mínimo de documentos de entrada e anotações necessários para treinar um modelo depende do tipo de anotações.

Anotações em PDF

Para criar um modelo para analisar arquivos de imagem ou documentos do Word, treine seu reconhecedor usando anotações em PDF. PDFs Para anotações em PDF, forneça pelo menos 250 documentos de entrada e pelo menos 100 anotações por entidade.

Se você fornecer um conjunto de dados de teste, os dados de teste devem incluir pelo menos uma anotação para cada um dos tipos de entidade especificados na solicitação de criação.

Anotações em texto simples

Para criar um modelo para analisar documentos de texto, você pode treinar seu reconhecedor usando anotações em texto simples.

Para anotações em texto simples, forneça pelo menos três documentos de entrada anotados e pelo menos 25 anotações por entidade. Se você fornecer menos de 50 anotações no total, o Amazon Comprehend reservará mais de 10% dos documentos de entrada para testar o modelo (a menos que você tenha fornecido um conjunto de dados de teste na solicitação de treinamento). Não se esqueça de que o tamanho mínimo do corpo do documento é de 5 KB.

Se sua entrada contiver apenas alguns documentos de treinamento, você poderá encontrar um erro dizendo que os dados de entrada de treinamento contêm poucos documentos que mencionam uma das entidades. Envie o trabalho novamente com documentos adicionais que mencionem a entidade.

Se você fornecer um conjunto de dados de teste, os dados de teste devem incluir pelo menos uma anotação para cada um dos tipos de entidade especificados na solicitação de criação.

Para ver um exemplo de como comparar um modelo com um pequeno conjunto de dados, consulte Amazon Comprehend anuncia limites mais baixos de anotação para reconhecimento de entidades personalizadas no site do blog. AWS

Práticas recomendadas de anotação

Há vários aspectos a levar em consideração para obter o melhor resultado ao usar anotações, inclusive:

- Anotar seus dados com cuidado e verifiqcar se você anotou todas as menções à entidade. Anotações imprecisas podem levar a resultados ruins.
- Os dados de entrada não devem conter duplicatas, como a duplicata de um PDF que você vai anotar. A presença de uma amostra duplicada pode resultar na contaminação do conjunto de testes e afetar negativamente o processo de treinamento, as métricas do modelo e o comportamento do modelo.
- Certifique-se de que todos os seus documentos estejam anotados e que os documentos sem anotações sejam decorrentes da falta de entidades legítimas e não por negligência. Por exemplo, se você tiver um documento que diz “J Doe é engenheiro há 14 anos”, você também deve fornecer uma anotação para “J Doe”, bem como para “John Doe”. A não observância disso confundirá o

modelo e poderá fazer com que o modelo não reconheça “J Doe” como ENGENHEIRO. Isso deve ser consistente no mesmo documento e em todos os documentos.

- Em geral, mais anotações levam a melhores resultados.
- Você pode treinar um modelo com o número mínimo de documentos e anotações, mas a adição de dados costuma melhorar o modelo. Recomendamos aumentar o volume de dados anotados em 10% para aumentar a precisão do modelo. Você pode executar a inferência em um conjunto de dados de teste que permaneça inalterado e possa ser testado por diferentes versões do modelo. Em seguida, você pode comparar as métricas de versões sucessivas do modelo.
- Forneça documentos que tenham a maior semelhança possível com os casos de uso reais. Deve-se evitar dados sintetizados com padrões repetitivos. Os dados de entrada devem ser os diversos possíveis para evitar sobreajustes e ajudar o modelo subjacente a fazer uma generalização melhor em exemplos reais.
- É importante que os documentos sejam diversos em termos de contagem de palavras. Por exemplo, se todos os documentos nos dados de treinamento forem curtos, o modelo resultante poderá ter dificuldade em prever entidades em documentos mais longos.
- Tente fornecer a mesma distribuição de dados para treinamento que você espera usar quando estiver detectando efetivamente suas entidades personalizadas (tempo de inferência). Por exemplo, no momento da inferência, se você espera nos enviar documentos que não contenham entidades, isso também deverá integrar seu conjunto de documentos de treinamento.

Para obter sugestões adicionais, consulte Como melhorar o desempenho do reconhecedor personalizado de entidades.

Arquivos de anotação em texto simples

Para anotações em texto simples, crie um arquivo de valores separados por vírgulas (CSV) contendo uma lista de anotações. O arquivo CSV deverá conter as seguintes colunas se o formato de entrada do arquivo de treinamento estiver configurado como um documento por linha.

Arquivo	Linha	Início deslocamento	Final deslocamento	Tipo
O nome do arquivo que contém o	O número da linha que contém a entidade.	O deslocamento de caracteres no texto de entrada	O deslocamento de caracteres no texto de entrada	O tipo de entidade definido pelo cliente. Os

Arquivo	Linha	Início deslocamento	Final deslocamento	Tipo
documento. Por exemplo, se um dos arquivos do documento estiver localizado em `s3://my-S3-bucket/test-files/documents.txt`, o valor na coluna `File` será `documents.txt`. Você deve incluir a extensão do arquivo (nesse caso, "`.txt`") como parte do nome do arquivo.	Omita essa coluna se o formato de entrada estiver configurado como um documento por arquivo.	(em relação ao início da linha) que mostra onde a entidade começa. O primeiro caractere está na posição 0.	que mostra onde a entidade termina.	tipos de entidade devem ser uma sequência de caracteres em letras maiúsculas separadas por sublinhado. Recomendamos usar tipos de entidade descritivos, como `MANAGER`, `SENIOR_MANAGER` ou `PRODUCT_CODE`. É possível treinar até 25 tipos de entidades por modelo.

Se o formato de entrada do arquivo de treinamento estiver configurado como um documento por arquivo, você omitirá a coluna do número de linha e os valores de Início deslocamento e Final deslocamento serão os deslocamentos da entidade desde o início do documento.

O exemplo a seguir é para um documento por linha. O arquivo `documents.txt` contém 4 linhas (linhas 0, 1, 2 e 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.
```

O arquivo CSV com a lista de anotações é o seguinte:

```
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**
>
> No arquivo de anotações, o número da linha que contém a entidade começa com a linha 0. Neste exemplo, o arquivo CSV não contém nenhuma entrada para a linha 2 porque não há nenhuma entidade na linha 2 de `documents.txt`.

Como criar seus arquivos de dados

Para reduzir o risco de erros, é importante colocar suas anotações em um arquivo CSV configurado corretamente. Para configurar seu arquivo CSV manualmente, o seguinte deve ser verdadeiro:

- A codificação UTF-8 deve ser especificada explicitamente, mesmo que seja usada como padrão na maioria dos casos.
- A primeira linha contém os cabeçalhos de coluna: `File`, `Line` (opcional), `Begin Offset`, `End Offset`, `Type`.

É altamente recomendável que você gere os arquivos de entrada CSV programaticamente para evitar possíveis problemas.

O exemplo a seguir usa Python para gerar um CSV para as anotações apresentadas anteriormente:

```
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"])
```

Arquivos de anotação em PDF

Para anotações em PDF, você usa o SageMaker AI Ground Truth para criar um conjunto de dados rotulado em um arquivo de manifesto aumentado. O Ground Truth é um serviço de rotulagem de dados que ajuda você (ou uma força de trabalho que você emprega) a criar conjuntos de dados de treinamento para modelos de machine learning. O Amazon Comprehend aceita arquivos de manifesto aumentados como dados de treinamento para modelos personalizados. Você pode fornecer esses arquivos ao criar um reconhecedor de entidades personalizado usando o console CreateEntityRecognizerdo Amazon Comprehend ou a ação da API.

Você pode usar o tipo de tarefa integrada Ground Truth, o Reconhecimento de entidade nomeada, para criar um trabalho de rotulagem para que os trabalhadores identifiquem entidades no texto. Para saber mais, consulte Reconhecimento de entidades nomeadas no Amazon SageMaker AI Developer Guide. Para saber mais sobre o Amazon SageMaker Ground Truth, consulte Usar o Amazon SageMaker AI Ground Truth para rotular dados.

> **Note**
>
> Usando o Ground Truth, você pode definir rótulos sobrepostos (texto que você associa a mais de um rótulo). No entanto, o reconhecimento de entidades do Amazon Comprehend não suporta rótulos sobrepostos.

Os arquivos de manifesto aumentado estão no formato de linhas JSON. Nesses arquivos, cada linha é um objeto JSON completo que contém um documento de treinamento e seus rótulos associados. O exemplo a seguir é um arquivo de manifesto aumentado que treina um reconhecedor de entidades para detectar as profissões de indivíduos mencionados no texto:

```
{"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"}}
```

Cada linha nesse arquivo de linhas JSON é um objeto JSON completo, no qual os atributos incluem o texto do documento, as anotações e outros metadados do Ground Truth. O exemplo a seguir é um único objeto JSON no arquivo de manifesto aumentado, mas está formatado para facilitar a leitura:

```
{
  "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"
  }
```

```
}
```

Neste exemplo, o atributo `source` fornece o texto do documento de treinamento, e o atributo `NamedEntityRecognitionDemo` fornece as anotações para as entidades no texto. O nome do atributo `NamedEntityRecognitionDemo` é arbitrário e você fornece um nome de sua escolha ao definir o trabalho de rotulagem no Ground Truth.

Neste exemplo, o atributo `NamedEntityRecognitionDemo` é o nome de atributo de rótulo, que é o atributo que fornece os rótulos que um funcionário do Ground Truth atribui aos dados de treinamento. Ao fornecer seus dados de treinamento ao Amazon Comprehend, você deve especificar um ou mais nomes de atributos de rótulo. O número de nomes de atributos que você especifica dependerá de o arquivo de manifesto aumentado ser a saída de um único trabalho de rotulagem ou de um trabalho de rotulagem em cadeia.

Se o arquivo for a saída de um único trabalho de rotulagem, especifique o nome de atributo de rótulo único que foi usado quando o trabalho foi criado no Ground Truth.

Se seu arquivo for a saída de um trabalho de rotulagem em cadeia, especifique o nome de atributo de rótulo etiqueta para um ou mais trabalhos na cadeia. Cada nome de atributo de rótulo fornece as anotações de uma tarefa individual. Você pode especificar até 5 desses atributos para arquivos de manifesto aumentado que são produzidos por trabalhos de rotulagem em cadeia.

Em um arquivo de manifesto aumentado, o nome de atributo de rótulo geralmente segue a chave `source`. Se o arquivo for a saída de um trabalho em cadeia, haverá vários nomes de atributos de rótulo. Ao fornecer seus dados de treinamento ao Amazon Comprehend, forneça somente os atributos que contenham anotações relevantes para o seu modelo. Não especifique os atributos que terminem com “-metadata”.

Para obter mais informações sobre trabalhos de etiquetagem em cadeia e para obter exemplos dos resultados que eles produzem, consulte Encadeamento de trabalhos de etiquetagem no Amazon SageMaker AI Developer Guide.

Anotação de arquivos PDF

Antes de fazer anotações em seu treinamento PDFs no SageMaker AI Ground Truth, preencha os seguintes pré-requisitos:

- Instalar o python3.8.x
- Instalar o jq

- Instale o AWS CLI

  Se você estiver usando a região us-east-1, poderá pular a instalação AWS da CLI porque ela já está instalada com seu ambiente Python. Nesse caso, você cria um ambiente virtual para usar o Python 3.8 no AWS Cloud9.
- Configurar suas credenciais da AWS
- Crie uma força de trabalho privada da SageMaker AI Ground Truth para apoiar a anotação

  Certifique-se de registrar o nome da equipe de trabalho que você escolher em sua nova força de trabalho privada, conforme você a utiliza durante a instalação.

Tópicos

- Configuração de seu ambiente
- Fazer upload de um PDF em um bucket do S3
- Criação de um trabalho de anotação
- Anotando com o SageMaker AI Ground Truth

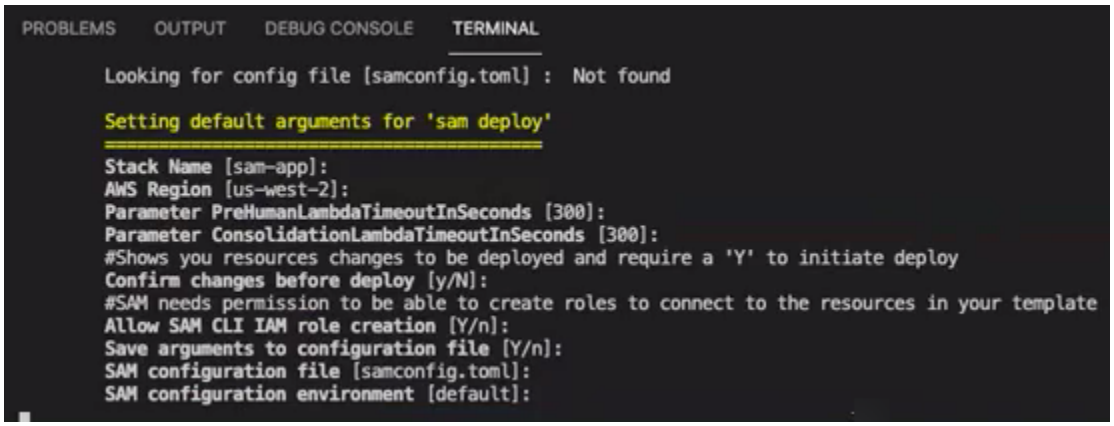
Configuração de seu ambiente

1. Se estiver usando o Windows, instale o Cygwin; se estiver usando Linux ou Mac, pule esta etapa.
2. Baixe os artefatos de anotação em. GitHub Descompacte o arquivo.
3. Na janela do terminal, navegue até a pasta descompactada (amazon-comprehend-semi-structured- documents-annotation-tools-main).
4. Essa pasta inclui uma opção `Makefiles` que você executa para instalar dependências, configurar um virtualenv do Python e implantar os recursos necessários. Revise o arquivo readme para fazer sua escolha.
5. A opção recomendada usa um único comando para instalar todas as dependências em um virtualenv, criar a AWS CloudFormation pilha a partir do modelo e implantá-la em você com orientação interativa. Conta da AWS Execute o seguinte comando:

   ```
   make ready-and-deploy-guided
   ```

   Esse comando apresenta um conjunto de opções de configuração. Verifique se o seu Região da AWS está correto. Para todos os outros campos, você pode aceitar os valores padrão ou

preencher valores personalizados. Se você modificar o nome da AWS CloudFormation pilha, anote-o conforme necessário nas próximas etapas.



A CloudFormation pilha cria e gerencia os AWS lambdas, as funções AWS do IAM e os buckets do AWS S3 necessários para a ferramenta de anotação.

Você pode revisar cada um desses recursos na página de detalhes da pilha no CloudFormation console.

6. O comando solicita que você inicie a implantação. CloudFormation cria todos os recursos na região especificada.

Quando o status da CloudFormation pilha muda para criação concluída, os recursos estão prontos para uso.

Fazer upload de um PDF em um bucket do S3

Na seção Configuração, você implantou uma CloudFormation pilha que cria um bucket S3 chamado comprehend-semi-structured-documents-$ {}. AWS::Region}-${AWS::AccountId Agora, você carrega seus documentos PDF de origem nesse bucket.

> **Note**
>
> Esse bucket contém os dados necessários para seu trabalho de marcação. A política do perfil de execução do Lambda concede permissão para que a função do Lambda acesse esse bucket.
> Você pode encontrar o nome do bucket S3 nos detalhes da CloudFormation pilha usando a chave SemiStructuredDocuments'S3Bucket'.

1. Crie uma nova pasta no bucket do S3. Nomeie essa nova pasta como “src”.
2. Adicione seus arquivos de origem PDF à sua pasta “src”. Em uma etapa posterior, você anotará esses arquivos para treinar seu reconhecedor.
3. (Opcional) Aqui está um exemplo de AWS CLI que você pode usar para carregar seus documentos de origem de um diretório local em um bucket do S3:

   ```
   aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided/src/
   ```

   Ou, com seu ID região da conta:

   ```
   aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided-Region-AccountID/src/
   ```
4. Agora você tem uma força de trabalho privada da SageMaker AI Ground Truth e enviou seus arquivos de origem para o bucket do S3, deploy-guided/src/; você está pronto para começar a anotar.

Criação de um trabalho de anotação

O script comprehend-ssie-annotation-tool-cli.py no `bin` diretório é um comando wrapper simples que simplifica a criação de um trabalho de rotulagem do SageMaker AI Ground Truth. O script do python lê os documentos de origem do seu bucket do S3 e cria um arquivo de manifesto de página única correspondente com um documento de origem por linha. Em seguida, o script cria um trabalho de rotulagem, que exige o arquivo de manifesto como entrada.

O script python usa o bucket e a CloudFormation pilha do S3 que você configurou na seção Configuração. Os parâmetros de entrada necessários para o script incluem:

- input-s3-path: Uri do S3 para os documentos de origem que você carregou no seu bucket do S3. Por exemplo: `s3://deploy-guided/src/`. Também é possível adicionar o ID e região da conta a esse caminho. Por exemplo: `s3://deploy-guided-Region-AccountID/src/`.
- cfn-name: O nome da CloudFormation pilha. Se você tiver usado o valor padrão para o nome da pilha, seu cfn-name será sam-app.
- work-team-name: o nome da força de trabalho que você criou ao criar a força de trabalho privada no SageMaker AI Ground Truth.
- job-name-prefix: O prefixo para o trabalho de rotulagem do SageMaker AI Ground Truth. Observe que há um limite de 29 caracteres para esse campo. Um carimbo de data/hora será anexado a esse valor. Por exemplo: `my-job-name-20210902T232116`.
- entity-types: as entidades que você deseja usar durante seu trabalho de rotulagem, separadas por vírgulas. Essa lista deve incluir todas as entidades que você deseja anotar no seu conjunto de dados de treinamento. A tarefa de rotulagem do Ground Truth exibe somente essas entidades para que os anotadores rotulem o conteúdo nos documentos PDF.

Para ver argumentos adicionais que o script suporta, use a opção `-h` para exibir o conteúdo da ajuda.

- Execute o script a seguir com os parâmetros de entrada, conforme descrito na lista anterior.

```
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"
```

O script produz o seguinte resultado:

```
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 %
```

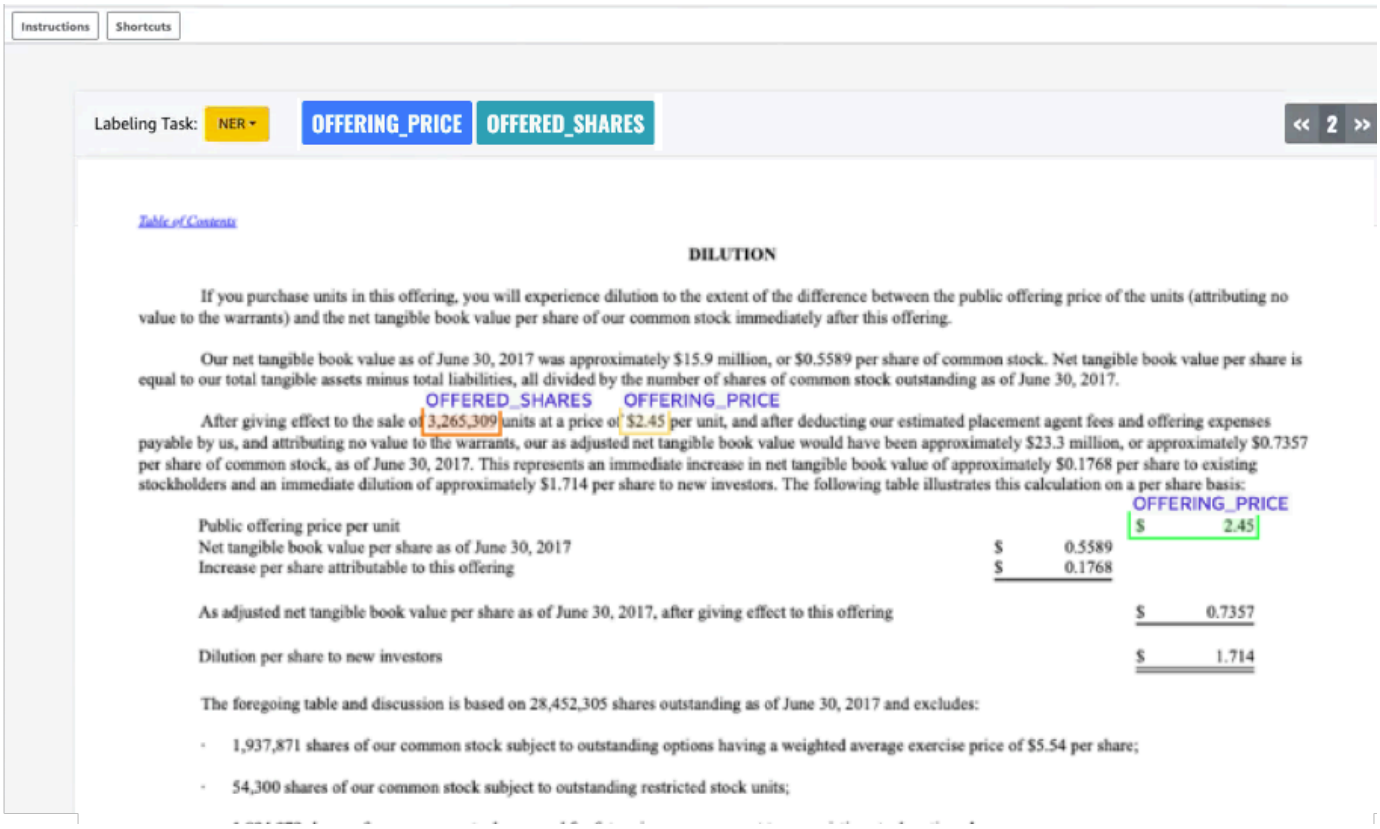
Anotando com o SageMaker AI Ground Truth

Agora que você configurou os recursos necessários e criou um trabalho de etiquetagem, você pode fazer login no portal de etiquetagem e anotar seu. PDFs

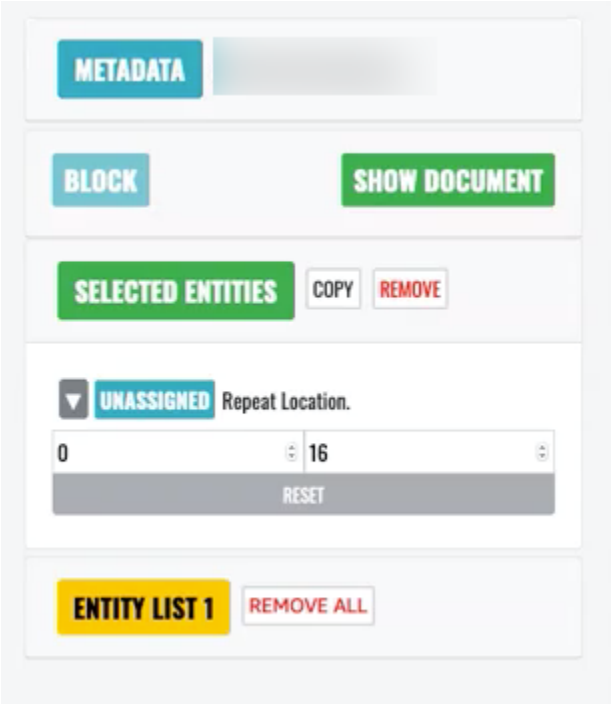
1. Faça login no console SageMaker AI usando os navegadores Chrome ou Firefox.
2. Selecione Rotulagem de forças de trabalho e escolha Privado.
3. Em Resumo da força de trabalho privada, selecione o URL de login do portal de rotulagem que você criou com sua força de trabalho privada. Faça login com as credenciais adequadas.

   Se você não encontrar nenhum trabalho listado, não se preocupe. A atualização pode demorar um pouco, dependendo do número de arquivos que você enviou para anotação.
4. Selecione sua tarefa e, no canto superior direito, escolha Começar a trabalhar para abrir a tela de anotação.

   Você verá um dos seus documentos abertos na tela de anotação e, acima dela, os tipos de entidade que você forneceu durante a configuração. À direita dos seus tipos de entidade, você verá uma seta que pode usar para navegar pelos seus documentos.



Faça anotações no documento aberto. Você também pode remover, desfazer ou marcar automaticamente suas anotações em cada documento. Essas opções estão disponíveis no painel direito da ferramenta de anotação.



Para usar a marcação automática, anote uma instância de uma de suas entidades e todas as outras instâncias dessa palavra específica serão então anotadas automaticamente com esse tipo de entidade.

Após terminar, selecione Enviar no canto inferior direito e use as setas de navegação para acessar o próximo documento. Repita isso até que você tenha anotado todos os seus. PDFs

Após anotar todos os documentos de treinamento, você poderá encontrar as anotações no formato JSON no bucket do Amazon S3 neste local:

```
/output/your labeling job name/annotations/
```

A pasta de saída também conterá um arquivo de manifesto de saída, que lista todas as anotações em seus documentos de treinamento. Você encontrará seu arquivo de manifesto de saída no seguinte local.

```
/output/your labeling job name/manifests/
```

Treinamento de modelos reconhecedores de entidades personalizados

Um reconhecedor personalizado de entidades identifica somente os tipos de entidade que você inclui ao treinar o modelo. Ele não inclui automaticamente os tipos de entidade predefinidos. Se você também quiser identificar os tipos de entidade predefinidos, como LOCALIZAÇÃO, DATA ou PESSOA, precisará fornecer dados adicionais de treinamento para essas entidades.

Ao criar um reconhecedor personalizado de entidades usando arquivos PDF anotados, você pode usar o reconhecedor com vários formatos de arquivo de entrada: texto simples, arquivos de imagem (JPG, PNG, TIFF), arquivos PDF e documentos do Word, sem a necessidade de pré-processamento ou nivelamento de documentos. O Amazon Comprehend não oferece suporte à anotação de arquivos de imagem ou documentos do Word.

> **Note**
>
> Um reconhecedor personalizado de entidades que use arquivos PDF anotados oferece suporte somente a documentos em inglês.

Depois de criar um reconhecedor de entidade personalizado, você pode monitorar o progresso da solicitação usando a DescribeEntityRecognizeroperação. Quando o campo `Status` for `TRAINED`, o modelo do reconhecedor estará pronto para ser usado no reconhecimento personalizado de entidades.

Tópicos

- Treinar reconhecedores personalizados (console)
- Treinar reconhecedores personalizados de entidades (API)
- Métricas personalizadas de reconhecimento de entidades

Treinar reconhecedores personalizados (console)

Crie reconhecedores de entidade personalizados usando o console do Amazon Comprehend. Esta seção mostra como criar e treinar um reconhecedor de entidade personalizado.

Criar de um reconhecedor de entidade personalizado usando o console: formato CSV

Para criar o reconhecedor de entidades personalizado, primeiro forneça um conjunto de dados para treinar seu modelo. Com esse conjunto de dados, inclua um dos seguintes: um conjunto de documentos anotados ou uma lista de entidades e seu rótulo de tipo, junto com um conjunto de documentos contendo essas entidades. Para ter mais informações, consulte Reconhecimento de entidade personalizado

Treinar um reconhecedor de entidade personalizado com um arquivo CSV

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Personalização e, em seguida, Reconhecimento personalizado de entidade.
3. Escolha Criar novo modelo.
4. Dê um nome ao reconhecedor. O nome deve ser exclusivo em uma região e conta.
5. Escolha o idioma.
6. Em Tipo de entidade personalizada, insira um rótulo personalizado que deseja que o reconhecedor encontre no conjunto de dados.

   O tipo de entidade deve estar em maiúsculas e, se consistir em mais de uma palavra, separe as palavras com um sublinhado.

7. Escolha Adicionar tipo.
8. Se quiser adicionar um tipo de entidade adicional, insira-o e escolha Adicionar tipo. Se quiser remover um dos tipos de entidade adicionados, escolha Remover tipo e selecione o tipo de entidade a ser removido da lista. Podem ser listados, no máximo, 25 tipos de entidades.
9. Para criptografar sua tarefa de treinamento, escolha Criptografia do reconhecedor e, em seguida, se deseja usar uma chave KMS associada à conta atual ou de outra conta.

   - Se estiver usando uma chave associada à conta atual, escolha o ID da chave KMS.
   - Se estiver usando uma chave associada a uma conta diferente, insira o ARN da chave KMS para o ID da chave.

   > **Note**
   >
   > Para mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte AWS Key Management Service.

10. Em Especificações de dados, escolha o formato dos seus documentos de treinamento:

    - Arquivo CSV: um arquivo CSV que complementa seus documentos de treinamento. O arquivo CSV contém informações sobre as entidades personalizadas que seu modelo treinado vai detectar. O formato exigido do arquivo depende se você está fornecendo anotações ou uma lista de entidades.
    - Manifesto aumentado — Um conjunto de dados rotulado produzido pela Amazon Ground Truth SageMaker . Esse arquivo está no formato de linhas JSON. Cada linha é um objeto JSON completo que contém um documento de treinamento e seus rótulos. Cada rótulo anota uma entidade nomeada no documento de treinamento. Você pode fornecer até 5 arquivos de manifesto aumentado.

    Para mais informações sobre os formatos disponíveis e para obter exemplos, consulte Treinamento de modelos reconhecedores de entidades personalizados.

11. Em Tipo de treinamento, escolha o tipo de treinamento a ser usado:

    - Utilizar anotações e documentos de treinamento
    - Utilizar lista de entidades e documentos de treinamento

Se escolher anotações, insira a URL do arquivo de anotações no Amazon S3. Você também pode navegar até o bucket ou pasta no Amazon S3 onde os arquivos de anotação estão localizados e escolher Procurar no S3.

Se escolher a lista de entidades, insira a URL da lista no Amazon S3. Você também pode navegar até o bucket ou pasta no Amazon S3 onde a lista de entidades está localizada e escolher Procurar no S3.

12. Insira a URL de um conjunto de dados de entrada contendo os documentos de treinamento no Amazon S3. Você também pode navegar até o bucket ou pasta no Amazon S3 onde os documentos de treinamento estão localizados e escolher Selecionar pasta.
13. Em Conjunto de dados de teste, selecione como deseja avaliar o desempenho do seu modelo treinado, tanto para anotações quanto para os tipos de treinamento da lista de entidades.

    - Divisão automática: a divisão automática seleciona automaticamente 10% dos seus dados de treinamento para usar como dados de teste.
    - (Opcional) Fornecido pelo cliente: ao selecionar essa opção, você pode especificar exatamente quais dados de teste deseja usar.
14. Se selecionar o conjunto de dados de teste fornecido pelo cliente, insira a URL do arquivo de anotações no Amazon S3. Você também pode navegar até o bucket ou pasta no Amazon S3 onde os arquivos de anotação estão localizados e escolher Selecionar pasta.
15. Na seção Escolha um perfil do IAM, selecione um perfil do IAM existente ou crie um novo.

    - Escolha um perfil do IAM existente: selecione essa opção se já tiver um perfil do IAM com permissões para acessar os buckets de entrada e saída do Amazon S3.
    - Criar um novo perfil do IAM: selecione essa opção quando quiser criar um novo perfil do IAM com as permissões adequadas para que o Amazon Comprehend acesse os buckets de entrada e saída.

    > **Note**
    >
    > Se os documentos de entrada forem criptografados, o perfil do IAM usado deverá ter permissão para o `kms:Decrypt`. Para obter mais informações, consulte Permissões necessárias para usar a criptografia KMS.

16. (Opcional) Para lançar seus recursos no Amazon Comprehend a partir de uma VPC, insira o ID da VPC em VPC ou escolha o ID na lista suspensa.

    1. Escolha a sub-rede em Sub-rede(s). Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.
    2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

    > **Note**
    >
    > Quando você usa uma VPC com sua tarefa personalizada de reconhecimento de entidades, o `DataAccessRole` usado para as operações Create and Start deve ter permissões para a VPC a partir da qual os documentos de entrada e o bucket de saída são acessados.

17. (Opcional) Para adicionar uma tag ao reconhecedor de entidade personalizado, insira um par de valores-chave em Tags. Escolha Adicionar Tag. Para remover esse par antes de criar o reconhecedor, escolha Remover tag.
18. Escolha Treinar.

O novo reconhecedor aparecerá na lista, mostrando seu status. Ele será exibido primeiro como `Submitted`. Em seguida, ele será exibido como `Training` para um classificador que está processando documentos de treinamento, `Trained` para um classificador pronto para uso e `In error` para um classificador com erro. Você pode clicar em uma tarefa para obter mais informações sobre o reconhecedor, incluindo mensagens de erro.

Criar de um reconhecedor de entidade personalizado usando o console: manifesto aumentado

Treinar um reconhecedor de entidade personalizado com um documento de texto sem formatação, PDF ou Word

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.
2. No menu à esquerda, escolha Personalização e, em seguida, Reconhecimento personalizado de entidade.
3. Escolha Treinar reconhecedor.
4. Dê um nome ao reconhecedor. O nome deve ser exclusivo em uma região e conta.

5. Escolha o idioma. Observação: se estiver treinando um documento PDF ou Word, o inglês é o idioma compatível.
6. Em Tipo de entidade personalizada, insira um rótulo personalizado que deseja que o reconhecedor encontre no conjunto de dados.

   O tipo de entidade deve estar em maiúsculas e, se consistir em mais de uma palavra, separe as palavras com um sublinhado.
7. Escolha Adicionar tipo.
8. Se quiser adicionar um tipo de entidade adicional, insira-o e escolha Adicionar tipo. Se quiser remover um dos tipos de entidade adicionados, escolha Remover tipo e selecione o tipo de entidade a ser removido da lista. Podem ser listados, no máximo, 25 tipos de entidades.
9. Para criptografar sua tarefa de treinamento, escolha Criptografia do reconhecedor e, em seguida, se deseja usar uma chave KMS associada à conta atual ou de outra conta.

   - Se estiver usando uma chave associada à conta atual, escolha o ID da chave KMS.
   - Se estiver usando uma chave associada a uma conta diferente, insira o ARN da chave KMS para o ID da chave.

   > **Note**
   >
   > Para mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte AWS Key Management Service.
10. Em Dados de treinamento, escolha Manifesto aumentado como seu formato de dados:

    - Manifesto aumentado — é um conjunto de dados rotulado produzido pela Amazon Ground Truth SageMaker . Esse arquivo está no formato de linhas JSON. Cada linha do arquivo é um objeto JSON completo contendo um documento de treinamento e seus rótulos. Cada rótulo anota uma entidade nomeada no documento de treinamento. Você pode fornecer até 5 arquivos de manifesto aumentado. Se estiver usando documentos PDF para dados de treinamento, deverá selecionar Manifesto aumentado. Você pode fornecer até 5 arquivos de manifesto aumentado. Para cada arquivo, nomeie até 5 atributos para usar como dados de treinamento.

    Para mais informações sobre os formatos disponíveis e para obter exemplos, consulte Treinamento de modelos reconhecedores de entidades personalizados.

11. Selecione o tipo de modelo de treinamento.

    Se você selecionou documentos de texto simples, em Local de entrada, insira o Amazon S3URL do arquivo de manifesto aumentado Amazon SageMaker AIGround Truth. Você também pode navegar até o bucket ou pasta no Amazon S3 onde os manifesto(s) aumentado(s) estão localizados e escolher Selecionar pasta.
12. Para os Nomes dos atributos, insira o nome do atributo contendo suas anotações. Se o arquivo contiver anotações de várias tarefas de rotulagem em cadeia, adicione um atributo para cada tarefa. Nesse caso, cada atributo contém o conjunto de anotações de uma tarefa de rotulagem. Observação: você pode fornecer até 5 nomes de atributos para cada arquivo.
13. Select Adicionar.
14. Se você selecionou documentos PDF e Word em Local de entrada, insira o Amazon S3URL do arquivo de manifesto aumentado do Amazon SageMaker AI Ground Truth. Você também pode navegar até o bucket ou pasta no Amazon S3 onde os manifesto(s) aumentado(s) estão localizados e escolher Selecionar pasta.
15. Insira o prefixo S3 para seus arquivos de dados de Anotação. Esses são os documentos PDF que você rotulou.
16. Insira o prefixo S3 para seus documentos de Origem. Esses são os documentos PDF originais (objetos de dados) fornecidos ao Ground Truth para sua tarefa de rotulagem.
17. Insira os nomes dos atributos que contêm suas anotações. Observação: você pode fornecer até 5 nomes de atributos para cada arquivo. Todos os atributos não especificados em seu arquivo serão ignorados.
18. Na seção de perfil do IAM, selecione um perfil existente do IAM ou crie um novo.

    - Escolha um perfil do IAM existente: selecione essa opção se já tiver um perfil do IAM com permissões para acessar os buckets de entrada e saída do Amazon S3.
    - Criar um novo perfil do IAM: selecione essa opção quando quiser criar um novo perfil do IAM com as permissões adequadas para que o Amazon Comprehend acesse os buckets de entrada e saída.

    > **Note**
    >
    > Se os documentos de entrada forem criptografados, o perfil do IAM usado deverá ter permissão para o `kms:Decrypt`. Para obter mais informações, consulte Permissões necessárias para usar a criptografia KMS.

19. (Opcional) Para lançar seus recursos no Amazon Comprehend a partir de uma VPC, insira o ID da VPC em VPC ou escolha o ID na lista suspensa.

    1. Escolha a sub-rede em Sub-rede(s). Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.
    2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

    > **Note**
    >
    > Quando você usa uma VPC com sua tarefa personalizada de reconhecimento de entidades, o `DataAccessRole` usado para as operações Create and Start deve ter permissões para a VPC a partir da qual os documentos de entrada e o bucket de saída são acessados.

20. (Opcional) Para adicionar uma tag ao reconhecedor de entidade personalizado, insira um par de valores-chave em Tags. Escolha Adicionar Tag. Para remover esse par antes de criar o reconhecedor, escolha Remover tag.
21. Escolha Treinar.

O novo reconhecedor aparecerá na lista, mostrando seu status. Ele será exibido primeiro como `Submitted`. Em seguida, ele será exibido como `Training` para um classificador que está processando documentos de treinamento, `Trained` para um classificador pronto para uso e `In error` para um classificador com erro. Você pode clicar em uma tarefa para obter mais informações sobre o reconhecedor, incluindo mensagens de erro.

Treinar reconhecedores personalizados de entidades (API)

Para criar e treinar um modelo personalizado de reconhecimento de entidades, use a operação da API Amazon CreateEntityRecognizerComprehend

Tópicos

- Treinar reconhecedores de entidades personalizados usando o AWS Command Line Interface
- Treinar reconhecedores de entidades personalizados usando o AWS SDK para Java
- Treinar reconhecedores personalizados de entidades usando Python (Boto3)

Treinar reconhecedores de entidades personalizados usando o AWS Command Line Interface

Os exemplos a seguir demonstram o uso da `CreateEntityRecognizer` operação e de outras APIs associadas à AWS CLI.

O exemplo está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Crie um reconhecedor personalizado de entidades usando o comando `create-entity-recognizer` da CLI. Para obter informações sobre o input-data-config parâmetro, consulte CreateEntityRecognizerna Referência da API Amazon Comprehend.

```
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
```

Liste todos os reconhecedores de entidades em uma região usando o comando `list-entity-recognizers` da CLI.

```
aws comprehend list-entity-recognizers \
     --region region
```

Verifique o status do trabalho dos reconhecedores personalizados de entidades usando o comando `describe-entity-recognizer` da CLI.

```
aws comprehend describe-entity-recognizer \
     --entity-recognizer-arn arn:aws:comprehend:region:account number:entity-
recognizer/test-6 \
     --region region
```

Treinar reconhecedores de entidades personalizados usando o AWS SDK para Java

Este exemplo cria um reconhecedor personalizado de entidade e treina o modelo usando Java

Para obter exemplos do Amazon Comprehend com Java, consulte exemplos em Java para o Amazon Comprehend.

Treinar reconhecedores personalizados de entidades usando Python (Boto3)

Instancie o SDK do Boto3:

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Crie um reconhecedor de entidades:

```
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"]
```

Listar todos os reconhecedores:

```
response = comprehend.list_entity_recognizers()
```

Aguarde até que o reconhecedor atinja o status TREINADO:

```
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)
```

Métricas personalizadas de reconhecimento de entidades

O Amazon Comprehend fornece métricas para ajudar você a estimar o quão bem um reconhecedor de entidades deve trabalhar em sua tarefa. Eles se baseiam no treinamento do modelo reconhecedor e, portanto, embora representem com precisão o desempenho do modelo durante o treinamento, são apenas uma aproximação do desempenho da API durante a descoberta da entidade.

As métricas são retornadas sempre que os metadados de um reconhecedor de entidades treinado são retornados.

O Amazon Comprehend é compatível com o treinamento de um modelo em até 25 entidades por vez. Quando as métricas são retornadas de um reconhecedor de entidade treinado, os scores são calculados em relação ao reconhecedor como um todo (métricas globais) e para cada entidade individual (métricas da entidade).

Três métricas estão disponíveis, tanto como métricas globais quanto de entidades:

- Precisão

  Isso indica a fração de entidades produzidas pelo sistema que estão corretamente identificadas e rotuladas corretamente. Isso mostra quantas vezes a identificação da entidade do modelo é realmente uma boa identificação. O número é uma porcentagem total da porcentagem.

  Em outras palavras, a precisão é baseada em verdadeiros positivos (tp) e falsos positivos (fp) e é calculada como precisão = tp/(tp + fp).

  Por exemplo, se um modelo prevê que dois exemplos de uma entidade estão em um documento, onde na verdade há apenas um, o resultado é um verdadeiro positivo e um falso positivo. Nesse caso, precisão = 1/(1 + 1). A precisão é de 50%, pois, das duas entidades identificadas pelo modelo, uma está correta.

- Recall

  Ela indica a fração de entidades presentes nos documentos que estão corretamente identificados e rotulados pelo sistema. Matematicamente, isso é definido em termos do número total de identificações corretas, positivos verdadeiros (tp) e identificações perdidas, falsos negativos (fn).

  É calculado como recall = tp/(tp + fn). Por exemplo, se um modelo identifica corretamente uma entidade, mas perde duas outras instâncias em que essa entidade está presente, o resultado é um verdadeiro positivo e dois falsos negativos. Nesse caso, lembre-se = 1/(1 + 2). O recall é de 33,33%, pois uma entidade está correta entre os três exemplos possíveis.

- F1 score

  Essa é uma combinação das métricas de precisão e recall, que mede a precisão geral do modelo para reconhecimento de entidades personalizadas. O F1 score é a média harmônica das métricas de precisão e recall: F1 = 2 * precisão * recall/(precisão + recall).

  > **Note**
  >
  > Intuitivamente, a média harmônica penaliza os extremos mais do que a média simples ou outros meios (exemplo: `precision` = 0, `recall` = 1 pode ser alcançado trivialmente ao prever todos os intervalos possíveis. Aqui, a média simples seria 0,5, mas o F1 penalizaria como 0).

  Nos exemplos acima, `precision` = 50% e `recall` = 33,33%, portanto F1 = 2 * 0,5 * 0,3333/(0,5 + 0,3333). O F1 score é 0,3975, ou 39,75%.

Métricas globais e individuais de entidades

A relação entre métricas de entidades globais e individuais pode ser vista ao analisar a frase a seguir para entidades que são um lugar ou uma pessoa.

```
John Washington and his friend Smith live in San Francisco, work in San Diego, and own
    a house in Seattle.
```

Em nosso exemplo, o modelo faz as previsões a seguir.

```
John Washington = Person
Smith = Place
San Francisco = Place
San Diego = Place
Seattle = Person
```

No entanto, as previsões deveriam ter sido as seguintes.

```
John Washington = Person
Smith = Person
San Francisco = Place
San Diego = Place
Seattle = Place
```

As métricas individuais da entidade para isso seriam:

```
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%
```

As métricas globais para isso seriam:

Global:

```
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))
```

Melhorar o desempenho do reconhecedor de entidade personalizado

Essas métricas fornecem um insight da precisão com que o modelo treinado funcionará quando você o usar para identificar entidades. Aqui estão algumas opções que usadas para melhorar suas métricas se elas estiverem abaixo das suas expectativas:

1. Dependendo se você usa Anotações ou Listas de entidades (somente texto simples), certifique-se de seguir as diretrizes na respectiva documentação para melhorar a qualidade dos dados. Se observar métricas melhores depois de melhorar seus dados e treinar novamente o modelo, poderá continuar iterando e melhorando a qualidade dos dados para um melhor desempenho.
2. Se estiver usando uma lista de entidades, considere usar anotações em vez disso. As anotações manuais geralmente podem melhorar seus resultados.
3. Se tiver certeza de que não há um problema de qualidade de dados e, ainda assim, as métricas permanecerem excessivamente baixas, envie uma solicitação de suporte.

Executar análise de reconhecedor personalizado em tempo real

A análise em tempo real é útil para aplicativos que processam pequenos documentos à medida que eles chegam. Por exemplo, você pode detectar entidades personalizadas em publicações de mídias sociais, tickets de suporte ou avaliações de clientes.

> **Antes de começar**
>
> Você precisa de um modelo de reconhecimento de entidade personalizado (também conhecido como reconhecedor) antes de poder detectar entidades personalizadas. Para mais informações sobre esses modelos, consulte the section called “Treinamento de modelos reconhecedores”.

Um reconhecedor treinado com anotações em texto sem formatação é compatível com a detecção de entidades somente para documentos em texto sem formatação. Um reconhecedor treinado com anotações de documentos PDF é compatível com a detecção de entidades para documentos de texto sem formatação, imagens, arquivos PDF e documentos do Word. Para informações sobre os arquivos de entrada, consulte Entradas para análise personalizada em tempo real.

Se você planeja analisar arquivos de imagem ou documentos PDF digitalizados, sua política do IAM deve conceder permissões para usar dois métodos da API Amazon Textract DetectDocumentText (AnalyzeDocumente). O Amazon Comprehend invoca esses métodos durante a extração de texto. Para visualizar um exemplo de política, consulte Permissões necessárias para executar ações de análise de documentos.

Tópicos

- Análise em tempo real para reconhecimento personalizado de entidades (console)
- Análise em tempo real para reconhecimento personalizado de entidades (API)
- Saídas para análise em tempo real

Análise em tempo real para reconhecimento personalizado de entidades (console)

Use o console do Amazon Comprehend para executar análises em tempo real com um modelo de personalizado. Primeiro, crie um endpoint para executar a análise em tempo real. Depois de criá-lo, execute a análise em tempo real.

Para informações sobre o fornecimento do throughput do endpoint e os custos associados, consulte Utilizar endpoints do Amazon Comprehend.

Tópicos

- Criação de um endpoint para detecção de entidades personalizadas

- Executar a detecção de entidades personalizadas em tempo real

Criação de um endpoint para detecção de entidades personalizadas

Criar um endpoint (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Endpoints e, depois, o botão Criar endpoint. A tela Criar endpoint é aberta.
3. Dê um nome ao endpoint. O nome deve ser exclusivo na sua conta e região atual.
4. Escolha um modelo personalizado ao qual você deseja anexar o novo endpoint. No menu suspenso, pesquise pelo nome do modelo.

   > **Note**
   >
   > Crie um modelo antes de poder anexar um endpoint a ele. Se ainda não tiver um modelo, consulte Treinamento de modelos reconhecedores de entidades personalizados.

5. (Opcional) Para adicionar uma tag ao endpoint, insira um par de valores-chave em Tags e escolha Adicionar tag. Para remover esse par antes de criar o endpoint, escolha Remover tag.
6. Insira o número de unidades de inferência (IUs) a serem atribuídas ao ponto final. Cada IU representa um throughput de 100 caracteres por segundo para até dois documentos por segundo. Para obter mais informações sobre o throughput do endpoint, consulte Utilizar endpoints do Amazon Comprehend.
7. (Opcional) Se você estiver criando um novo endpoint, terá a opção de usar o estimador de IU. O estimador pode ajudá-lo a determinar o número de solicitações a IUs serem solicitadas. O número de unidades de inferência depende da throughput ou do número de caracteres que você deseja analisar por segundo.
8. Em Resumo da compra, analise seu custo estimado de endpoint por hora, dia e mês.
9. Marque a caixa de seleção se você está ciente de que sua conta acumula cobranças pelo endpoint desde o momento em que ele é iniciado até que você o exclua.
10. Escolha Criar endpoint.

Executar a detecção de entidades personalizadas em tempo real

Depois de criar um endpoint para seu modelo personalizado de reconhecimento de entidades, você pode executar análises em tempo real para detectar entidades em documentos individuais.

Siga as etapas abaixo para detectar entidades personalizadas em seu texto usando o console do Amazon Comprehend.

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Análise em tempo real.
3. Na seção Texto de entrada, em Tipo de análise, escolha Personalizado.
4. Em Selecionar endpoint, escolha o endpoint associado ao modelo de detecção de entidades que você deseja usar.
5. Para especificar os dados de entrada para análise, insira um texto ou carregue um arquivo.
   - Inserir texto:
     a. Escolha Texto de entrada.
     b. Insira o texto que deseja analisar.
   - Como carregar um arquivo:
     a. Escolha Carregar arquivo e insira o nome do arquivo para carregar.
     b. (Opcional) Em Ações de leitura avançadas, você pode substituir as ações padrão para extração de texto. Para obter detalhes, consulte Configurar opções de extração de texto.
6. Escolha Analisar. O console exibe o resultado da análise, junto com uma avaliação de confiança.

Análise em tempo real para reconhecimento personalizado de entidades (API)

Você pode usar a API do Amazon Comprehend para executar a análise em tempo real com um modelo personalizado. Primeiro, crie um endpoint para executar a análise em tempo real. Depois de criá-lo, execute a análise em tempo real.

Para informações sobre o fornecimento do throughput do endpoint e os custos associados, consulte Utilizar endpoints do Amazon Comprehend.

Tópicos

- Criação de um endpoint para detecção de entidades personalizadas
- Executar a detecção de entidades personalizadas em tempo real

Criação de um endpoint para detecção de entidades personalizadas

Para obter informações sobre os custos associados aos endpoints, consulte Utilizar endpoints do Amazon Comprehend.

Criando um endpoint com o AWS CLI

Para criar um endpoint usando o AWS CLI, use o `create-endpoint` comando:

```
$ 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
```

Se seu comando for executado com êxito, o Amazon Comprehend responderá com o ARN do endpoint:

```
{
   "EndpointArn": "Arn"
}
```

Para obter mais informações sobre esse comando, seus argumentos de parâmetro e seu resultado consulte `create-endpoint` na Referência de comandos da AWS CLI .

Executar a detecção de entidades personalizadas em tempo real

Depois de criar um endpoint para seu modelo personalizado de reconhecimento de entidades, você usa o endpoint para executar a operação da DetectEntitiesAPI. Você pode fornecer entrada de texto usando os parâmetros`text` ou `bytes`. Insira os outros tipos de entrada usando o parâmetro `bytes`.

Para arquivos de imagem e arquivos PDF, você pode usar o parâmetro `DocumentReaderConfig` para substituir as ações padrão de extração de texto. Para obter detalhes, consulte Configurar opções de extração de texto.

Detectando entidades no texto usando o AWS CLI

Para detectar entidades personalizadas no texto, execute o comando `detect-entities` com o texto de entrada no parâmetro `text`.

Example : use a CLI para detectar entidades no texto de entrada

```
$ aws comprehend detect-entities \
> --endpoint-arn arn \
> --language-code en \
> --text  "Andy Jassy is the CEO of Amazon."
```

Se seu comando for executado com êxito, o Amazon Comprehend responderá com a análise. Para cada entidade que o Amazon Comprehend detecta, ele fornece o tipo de entidade, o texto, a localização e a pontuação de confiança.

Detectando entidades em documentos semiestruturados usando o AWS CLI

Para detectar entidades personalizadas de um arquivo em PDF, Word ou de imagem, execute o comando `detect-entities` com o arquivo de entrada no parâmetro `bytes`.

Example : use a CLI para detectar entidades em um arquivo de imagem

Este exemplo mostra como passar o arquivo de imagem usando a opção `fileb` para codificar os bytes da imagem em base64. Para obter mais informações, consulte Objetos binários grandes no Guia AWS Command Line Interface do usuário.

Esse exemplo também passa um arquivo JSON chamado `config.json` para definir as opções de extração de texto.

```
$ aws comprehend detect-entities \
> --endpoint-arn arn \
> --language-code en \
> --bytes fileb://image1.jpg   \
> --document-reader-config file://config.json
```

O arquivo config.json contém o conteúdo a seguir.

```
{
   "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
   "DocumentReadAction": "TEXTRACT_DETECT_DOCUMENT_TEXT"
}
```

Para obter mais informações sobre a sintaxe do comando, consulte a DetectEntitiesReferência da API Amazon Comprehend.

Saídas para análise em tempo real

Saídas para entradas de texto

Se você inserir texto usando o parâmetro `Text`, a saída consistirá em uma matriz de entidades que a análise detectou. O exemplo a seguir mostra uma análise que detectou duas entidades do 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"
            }
        ]
    }
```

Saídas para entradas semiestruturadas

Para um documento de entrada semiestruturado ou um arquivo de texto, a saída pode incluir os seguintes campos adicionais:

- DocumentMetadata — Extração de informações sobre o documento. Os metadados incluem uma lista de páginas no documento, com o número de caracteres extraídos de cada página. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- DocumentType — O tipo de documento para cada página no documento de entrada. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`.
- Blocks: informações sobre cada bloco de texto no documento de entrada. Os blocos estão aninhados. Um bloco de página contém um bloco para cada linha de texto, que contém um bloco para cada palavra. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`.
- BlockReferences — Uma referência a cada bloco dessa entidade. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`. O campo não está presente para arquivos de texto.
- Erros: erros no nível da página que o sistema detectou ao processar o documento de entrada. O campo estará vazio se o sistema não encontrar erros.

Para obter descrições desses campos de saída, consulte DetectEntitiesna Referência da API Amazon Comprehend. Para obter mais informações sobre os elementos do layout, consulte Objetos de análise do Amazon Textract no Guia do desenvolvedor do Amazon Textract.

O exemplo a seguir mostra a saída para um documento de entrada de uma página em PDF digitalizado.

```
{
    "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"
        }]
    }]
}
```

O exemplo a seguir mostra o resultado da análise de um documento em PDF nativo.

Example Exemplo de resultado de uma análise de reconhecimento personalizado de entidade de um documento em 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"
}
```

Executar trabalhos de análise para reconhecimento personalizado de entidades

Você pode executar um trabalho de análise assíncrona para detectar entidades personalizadas em um conjunto de um ou mais documentos.

> **Antes de começar**
>
> Você precisa de um modelo de reconhecimento de entidade personalizado (também conhecido como reconhecedor) antes de poder detectar entidades personalizadas. Para mais informações sobre esses modelos, consulte the section called “Treinamento de modelos reconhecedores”.

Um reconhecedor treinado com anotações em texto sem formatação é compatível com a detecção de entidades somente para documentos em texto sem formatação. Um reconhecedor treinado com anotações de documentos PDF é compatível com a detecção de entidades para documentos de texto sem formatação, imagens, arquivos PDF e documentos do Word. Para arquivos que não sejam arquivos de texto, o Amazon Comprehend executa a extração de texto antes de executar a análise. Para informações sobre os arquivos de entrada, consulte Entradas para análise personalizada assíncrona.

Se você planeja analisar arquivos de imagem ou documentos PDF digitalizados, sua política do IAM deve conceder permissões para usar dois métodos da API Amazon Textract DetectDocumentText (AnalyzeDocumente). O Amazon Comprehend invoca esses métodos durante a extração de texto. Para visualizar um exemplo de política, consulte Permissões necessárias para executar ações de análise de documentos.

Para executar um trabalho de análise assíncrona, execute as seguintes etapas gerais:

1. Armazene os documentos em um bucket do Amazon S3.
2. Use a API ou o console para iniciar o trabalho de análise.
3. Monitore o progresso do trabalho de análise.
4. Depois que o trabalho for concluído, recupere os resultados da análise do bucket do S3 que você especificou ao iniciar o trabalho.

Tópicos

- Iniciar um trabalho de detecção personalizada de entidades (console)
- Iniciar um trabalho personalizado de detecção de entidades (API)
- Saídas para trabalhos de análise assíncrona

Iniciar um trabalho de detecção personalizada de entidades (console)

Você pode usar o console para iniciar e monitorar um trabalho de análise assíncrona para reconhecimento personalizado de entidades.

Para iniciar um trabalho de análise assíncrona

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Tarefas de análise e, em seguida, Criar tarefa.
3. Dê um nome ao trabalho de classificação. O nome deve ser exclusivo à sua conta e na sua região atual.
4. Em Tipo de análise, escolha Reconhecimento personalizado de entidade.
5. De Reconhecedor de modelos, escolha o reconhecedor personalizado de entidade a ser usado.
6. De Versão, escolha a versão do reconhecedor a ser usada.
7. (Opcional) Se você optar por criptografar os dados que o Amazon Comprehend usa ao processar sua tarefa, escolha Criptografia de tarefas. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta.
   - Se estiver usando uma chave associada à conta atual, escolha o ID da chave para o ID da chave KMS.
   - Se estiver usando uma chave associada a uma conta diferente, insira o ARN do ID da chave em ARN da chave KMS.

   > **Note**
   >
   > Para obter mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte Serviço de gerenciamento de chaves (KMS).
8. Em Dados de entrada, insira a localização do bucket do Amazon S3 contendo seus documentos de entrada ou navegue até ele escolhendo Procurar no S3. Esse bucket deve estar na mesma

região da API que você está chamando. O perfil do IAM que você está usando para acessar as permissões de acesso do trabalho de análise deve ter permissões de leitura para o bucket do S3.

9. (Opcional) para Formato de entrada, escolha o formato dos documentos de entrada. O formato pode ser um documento por arquivo ou um documento por linha em um único arquivo. Um documento por linha se aplica somente a documentos de texto.
10. (Opcional) Para Modo de leitura do documento, você pode substituir as ações padrão de extração de texto. Para obter mais informações, consulte Configurar opções de extração de texto.
11. Em Dados de saída, insira a localização do bucket do Amazon S3 em que o Amazon Comprehend deve gravar os dados de saída da tarefa ou navegar até elas escolhendo Procurar no S3 Esse bucket deve estar na mesma região da API que você está chamando. O perfil do IAM que você está usando para obter permissões de acesso à tarefa de classificação deve ter permissões de gravação para o bucket do S3.
12. (Opcional) Se você optar por criptografar o resultado de saída da sua tarefa, escolha Criptografia. Em seguida, escolha se deseja usar uma chave KMS associada à conta atual ou uma de outra conta.

    - Se você estiver usando uma chave associada à conta atual, escolha o alias ou ID da chave para o ID da chave KMS.
    - Se você estiver usando uma chave associada a uma conta diferente, insira o ARN do alias ou ID da chave em ID da chave KMS.
13. (Opcional) Para lançar seus recursos no Amazon Comprehend a partir de uma VPC, insira o ID da VPC em VPC ou escolha o ID na lista suspensa.

    1. Escolha a sub-rede em Sub-rede(s). Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.
    2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

> **Note**
>
> Quando você usa uma VPC com o seu trabalho de análise, a `DataAccessRole` usada para as operações Criar e Iniciar deve ter permissões à VPC que acessa o bucket de saída.

14. Escolha Criar trabalho para criar o trabalho de reconhecimento de entidades.

Iniciar um trabalho personalizado de detecção de entidades (API)

Você pode usar a API para iniciar e monitorar um trabalho de análise assíncrona para reconhecimento personalizado de entidades.

Para iniciar um trabalho personalizado de detecção de entidades com a StartEntitiesDetectionJoboperação, você fornece o EntityRecognizerArn, que é o Amazon Resource Name (ARN) do modelo treinado. Você pode encontrar esse ARN na resposta à CreateEntityRecognizeroperação.

Tópicos

- Detectando entidades personalizadas usando o AWS Command Line Interface
- Detectando entidades personalizadas usando o AWS SDK para Java
- Detectando entidades personalizadas usando o AWS SDK para Python (Boto3)
- Substituindo ações de API para arquivos em PDF

Detectando entidades personalizadas usando o AWS Command Line Interface

Use o exemplo a seguir para ambientes Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^). Para detectar entidades personalizadas em um conjunto de documentos, use a seguinte sintaxe de solicitação:

```
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
```

O Amazon Comprehend responde com `JobID` e `JobStatus` e retornará a saída do trabalho no bucket do S3 que você especificou na solicitação.

Detectando entidades personalizadas usando o AWS SDK para Java

Para obter exemplos do Amazon Comprehend com Java, consulte exemplos em Java para o Amazon Comprehend.

Detectando entidades personalizadas usando o AWS SDK para Python (Boto3)

Este exemplo cria um reconhecedor de entidades personalizado, treina o modelo e o executa em um trabalho de reconhecimento de entidades usando o AWS SDK para Python (Boto3).

Instancie o SDK Python.

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Crie um reconhecedor de entidades:

```
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"]
```

Listar todos os reconhecedores:

```
response = comprehend.list_entity_recognizers()
```

Aguarde até que o reconhecedor de entidades atinja o status 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)
```

Inicie um trabalho personalizado de detecção de entidades:

```
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"
    }
)
```

Substituindo ações de API para arquivos em PDF

Para arquivos de imagem e arquivos em PDF, você pode substituir as ações padrão de extração usando o parâmetro `DocumentReaderConfig` em `InputDataConfig`.

O exemplo a seguir define um arquivo JSON chamado myInputData config.json para definir os valores. `InputDataConfig` Ele configura `DocumentReadConfig` para usar a API `DetectDocumentText` do Amazon Textract para todos os arquivos em 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"
  }
}
```

Na `StartEntitiesDetectionJob` operação, especifique o arquivo myInputData config.json como parâmetro: `InputDataConfig`

```
  --input-data-config file://myInputDataConfig.json
```

Para obter mais informações sobre o parâmetro `DocumentReaderConfig`, consulte Configurar opções de extração de texto.

Saídas para trabalhos de análise assíncrona

Depois que uma tarefa de análise é concluída, ela armazena os resultados no bucket do S3 que você especificou na solicitação.

Saídas para entradas de texto

Para arquivos de entrada de texto, a saída consiste em uma lista de entidades para cada documento de entrada.

O exemplo a seguir mostra a saída de dois documentos de um arquivo de entrada chamado 50_docs, usando um formato de documento por linha.

```
{
        "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"
            }
        ]
    }
```

Saídas para entradas semiestruturadas

Para documentos de entrada semiestruturados, a saída pode incluir os seguintes campos adicionais:

- DocumentMetadata — Extração de informações sobre o documento. Os metadados incluem uma lista de páginas no documento, com o número de caracteres extraídos de cada página. Esse campo estará presente na resposta se a solicitação incluir o parâmetro `Byte`.
- DocumentType — O tipo de documento para cada página no documento de entrada. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`.
- Blocks: informações sobre cada bloco de texto no documento de entrada. Os blocos podem se aninhar dentro de um bloco. Um bloco de página contém um bloco para cada linha de texto, que contém um bloco para cada palavra. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`.
- BlockReferences — Uma referência a cada bloco dessa entidade. Esse campo estará presente na resposta de uma solicitação que incluiu o parâmetro `Byte`. O campo não está presente para arquivos de texto.
- Erros: erros no nível da página que o sistema detectou ao processar o documento de entrada. O campo estará vazio se o sistema não encontrar erros.

Para obter mais detalhes sobre esses campos de saída, consulte a DetectEntitiesReferência da API Amazon Comprehend

O exemplo a seguir mostra a saída para um documento de entrada PDF nativo de uma página.

Example Exemplo de resultado de uma análise de reconhecimento personalizado de entidade de um documento em 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"
}
```

Criação e gerenciamento de modelos personalizados

O Amazon Comprehend inclui modelos integrados de PNL (processamento de linguagem natural) que você pode usar para analisar insights ou modelos de tópicos. Você pode usar o Amazon Comprehend para criar modelos personalizados para reconhecimento de entidades e classificação de documentos.

Você pode usar o versionamento do modelo para monitorar o histórico do seu modelo. Ao criar e treinar uma nova versão do modelo, você pode fazer alterações no conjunto de dados de treinamento. O Amazon Comprehend exibe detalhes (incluindo o desempenho do modelo) de cada versão do modelo na página de detalhes do modelo. Com o tempo, você pode ver como o desempenho do modelo muda à medida que você faz alterações no seu conjunto de dados de treinamento.

Você pode criar versões de modelo usando o console ou a API do Amazon Comprehend. Como alternativa, o Amazon Comprehend fornece Flywheels para simplificar as tarefas associadas ao treinamento e à avaliação de novas versões de modelos personalizados.

Depois de criar um modelo personalizado, você pode compartilhar o modelo com outros usuários permitindo que outros Contas da AWS importem uma cópia do seu modelo.

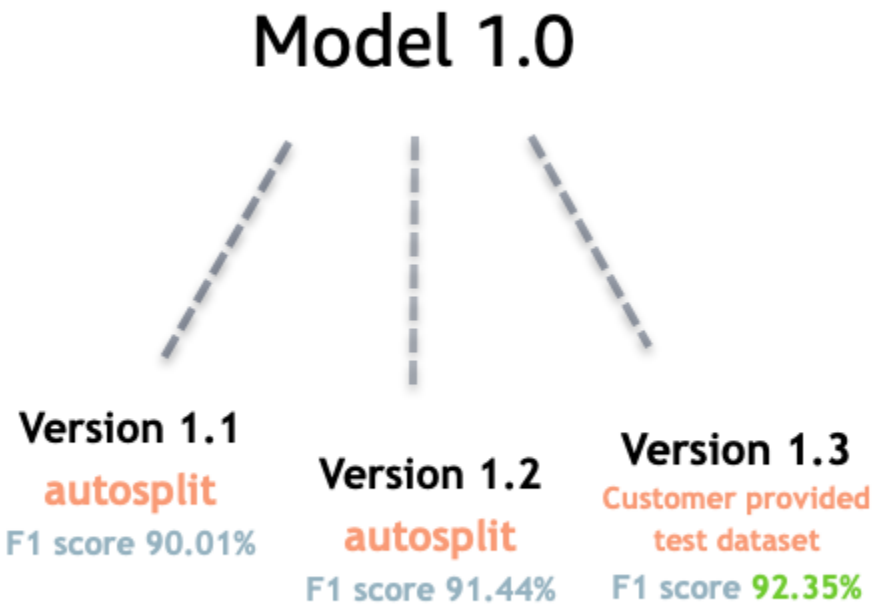
Tópicos

- Versionamento do modelo com o Amazon Comprehend
- Copiar modelos personalizados entre Contas da AWS

Versionamento do modelo com o Amazon Comprehend

Inteligência artificial e machine learning (IA/ML) têm tudo a ver com experimentação rápida. Com o Amazon Comprehend, você treina e compila modelos que você pode usar para obter insights sobre seus dados. Com o versionamento do modelo, você pode acompanhar seu histórico de modelos e pontuações associadas aos resultados em execução de seus modelos à medida que fornece mais ou diferentes conjuntos de dados. Você pode usar o versionamento com seus modelos de classificação personalizados ou seus modelos personalizados de reconhecimento de entidades. Analisando suas diferentes versões ao longo do tempo, você pode obter uma visão sobre o desempenho delas e sobre quais parâmetros você usou para chegar ao sucesso.

Ao treinar uma nova versão de um modelo de classificador personalizado existente ou de um modelo de reconhecimento de entidade, tudo o que você precisa fazer é criar uma nova versão na página de detalhes do modelo e todos os detalhes serão preenchidos para você. A nova versão terá o mesmo nome do seu modelo anterior, que chamamos de VersionID, embora você dê a ela um nome exclusivo durante a criação. Ao adicionar novas versões a um modelo, você pode ver todas as versões anteriores e seus detalhes em uma visualização na página de detalhes do modelo. Com o versionamento, você pode ver como a performance do modelo muda à medida que você faz alterações no conjunto de dados de treinamento.



Crie uma nova versão do classificador personalizado (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Personalização e, em seguida, Classificação personalizada.
3. Na lista Classificadores, escolha o nome do modelo personalizado do qual você deseja criar uma nova versão. A página de detalhes do modelo personalizado é exibida.
4. No canto superior direito, selecione Criar novo modelo. Uma tela é exibida com detalhes pré-preenchidos do modelo pai de classificação personalizado.

5. Em Nome da versão, adicione um nome exclusivo à nova versão.
6. Em detalhes da versão, você pode alterar o idioma e o número de rótulos associados ao seu novo modelo.
7. Na seção Especificações de dados, configure como você deseja fornecer os dados para sua nova versão. Verifique se você está fornecendo dados completos, que incluem documentos do seu modelo anterior e seus novos documentos. Você pode alterar o Modo do classificador (rótulo único ou rótulo múltiplo), o Formato dos dados (arquivo CSV, manifesto aumentado), seu Conjunto de dados de treinamento e seu Conjunto de dados de teste (divisão automática ou sua configuração de dados de teste personalizada).
8. (Opcional) Atualize a localização do S3 para seus dados de saída
9. Em Permissões de acesso, crie ou use um perfil do IAM existente.
10. (Opcional) Atualize suas configurações da VPC
11. (Opcional) Adicione tags à nova versão para ajudar a monitorar os detalhes.

    Para obter mais informações sobre como criar classificadores personalizados, consulte Crie uma classificador personalizado

Crie uma nova versão do reconhecedor de entidades personalizadas (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Personalização e, em seguida, Reconhecimento personalizado de entidade.
3. Na lista do Modelo do reconhecedor, escolha o nome do reconhecedor do qual você deseja criar uma nova versão. A página de detalhes será exibida.
4. No canto superior direito, selecione Treinar nova versão. Uma tela é aberta com detalhes pré-preenchidos do reconhecedor da entidade principal.
5. Em Nome da versão, adicione um nome exclusivo à nova versão.
6. Em Tipo personalizado da entidade, adicione os rótulos personalizados ou o rótulo que você deseja que o reconhecedor identifique no seu conjunto de dados e selecione Adicionar tipos. Escolha um tipo de entidade personalizado nas anotações ou na lista de entidades que você forneceu. O reconhecedor, então, usará todos os tipos de entidade incluídos para identificar entidades no conjunto de dados ao executar seu trabalho. Cada tipo de entidade deve estar em letras maiúsculas e separado por um sublinhado se usar várias palavras. São permitidos 25 tipos no máximo.

7. (Opcional) Selecione Criptografia do reconhecedor para criptografar os dados no volume de armazenamento enquanto seu trabalho está sendo processado.
8. Na seção Dados de treinamento, especifique os detalhes em Formato da anotação e dos dados (arquivo CSV, manifesto aumentado, rótulo único ou múltiplo), Formato de dados (CSV, manifesto aumentado), seu Conjunto de dados de treinamento e seu Conjunto de dados de teste (divisão automática ou configuração personalizada de dados de teste).
9. (Opcional) Atualize a localização do S3 para seus dados de saída
10. Em Permissões de acesso, crie ou use um perfil do IAM existente.
11. (Opcional) Atualize suas configurações da VPC
12. (Opcional) Adicione tags à nova versão para ajudar a monitorar os detalhes.

Para saber mais sobre reconhecedores de entidades personalizados, consulte Reconhecimento personalizado de entidades e Criar um reconhecedor de entidades personalizado usando o console.

Copiar modelos personalizados entre Contas da AWS

Os usuários do Amazon Comprehend podem copiar modelos personalizados treinados em um processo de Contas da AWS duas etapas. Primeiro, um usuário em uma Conta da AWS (conta A) compartilha um modelo personalizado que está em sua conta. Em seguida, um usuário em outra Conta da AWS (conta B) importa o modelo para sua conta. O usuário da conta B não precisa treinar o modelo e não precisa copiar (ou acessar) os dados originais de treinamento nem os dados de teste.

Para compartilhar um modelo personalizado na conta A, o usuário anexa uma política AWS Identity and Access Management (IAM) a uma versão do modelo. Essa política autoriza uma entidade na conta B, como um usuário ou perfil, a importar a versão do modelo para o Amazon Comprehend em sua Conta da AWS. O usuário da conta B deve importar o modelo para a mesma Região da AWS modelo original.

Para importar o modelo na conta B, o usuário dessa conta fornece os detalhes necessários ao Amazon Comprehend, como o nome do recurso da Amazon (ARN) do modelo. Ao importar o modelo, esse usuário cria um novo modelo personalizado Conta da AWS que replica o modelo que ele importou. Esse modelo é totalmente treinado e pronto para trabalhos de inferência, como classificação de documentos ou reconhecimento de entidades nomeadas.

Copiando um modelo personalizado é útil se:

- Você pertence a uma organização que usa vários Contas da AWS. Por exemplo, sua organização pode ter um Conta da AWS para cada fase de desenvolvimento, como criação, preparação, teste e implantação. Ou pode ter funções comerciais distintas Contas da AWS , como ciência e engenharia de dados.
- Sua organização trabalha com outra, como um AWS parceiro, que treina modelos personalizados no Amazon Comprehend e os fornece a você como cliente.

Em cenários como esses, você pode copiar rapidamente um reconhecedor de entidades personalizado ou classificador de documentos treinado de um Conta da AWS para outro. Copiar um modelo dessa forma é mais fácil do que a alternativa, na qual você copia dados de treinamento Contas da AWS para treinar modelos duplicados.

Tópicos

- Compartilhando um modelo personalizado com outro Conta da AWS
- Importando um modelo personalizado de outro Conta da AWS

Compartilhando um modelo personalizado com outro Conta da AWS

Com o Amazon Comprehend, você pode compartilhar seus modelos personalizados com outras pessoas, permitindo que elas importem seus modelos para as contas da AWS delas. Quando um usuário importa um de seus modelos personalizados, ele cria um novo modelo personalizado na conta dele. O novo modelo do usuário duplica o modelo que você compartilhou.

Para compartilhar um modelo personalizado, você anexa uma política que autorize outras pessoas a importá-lo. Em seguida, você fornece os detalhes necessários a esses usuários.

> **Note**
>
> Quando outros usuários importam um modelo personalizado que você compartilhou, eles devem usar o mesmo Região da AWS — por exemplo, Leste dos EUA (Norte da Virgínia) — que contém seu modelo.

Tópicos

- Antes de começar
- Políticas baseadas em recursos para modelos personalizados

- Etapa 1: incluir uma política baseada em recursos em um modelo personalizado
- Etapa 2: fornecer os detalhes que outras pessoas precisam importar

Antes de começar

Antes de compartilhar um modelo, você deve ter um classificador personalizado treinado ou um reconhecedor personalizado de entidade principal no Amazon Comprehend em sua Conta da AWS. Para mais informações sobre o treinamento de modelos personalizados, consulte Classificação personalizada ou Reconhecimento de entidade personalizado.

Permissões obrigatórias

Declaração de política do IAM

Antes que seja possível adicionar uma política baseada em recursos a um modelo personalizado, você precisa de permissões no AWS Identity and Access Management (IAM). Seu usuário, grupo ou perfil deve ter uma política anexada para que você possa criar, obter e excluir políticas de modelo, conforme apresentado no exemplo a seguir.

Example Política do IAM para gerenciar políticas baseadas em recursos para modelos personalizados

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:PutResourcePolicy",
    "comprehend:DeleteResourcePolicy",
    "comprehend:DescribeResourcePolicy"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Para obter mais informações sobre como criar uma política do IAM, consulte Criar políticas do IAM no Guia do usuário do IAM. Para obter informações sobre como anexar a política do IAM, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

AWS KMS declaração de política chave

Se você estiver compartilhando um modelo criptografado, talvez seja necessário adicionar permissões para AWS KMS o. Esse requisito dependerá do tipo de chave do KMS que você usa para criptografar o modelo no Amazon Comprehend.

An pertence e Chave pertencente à AWSé gerenciado por um AWS serviço. Se você usa um Chave pertencente à AWS, não precisa adicionar permissões para AWS KMS, e você pode pular esta seção.

Uma Chave gerenciada pelo cliente é uma chave que você cria, possui e gerencia na sua Conta da AWS. Se usar uma chave gerenciada pelo cliente, você deverá adicionar uma instrução à sua política de chave do KMS.

A declaração de política autoriza uma ou mais entidades (como usuários ou contas) a realizar as AWS KMS operações necessárias para descriptografar o modelo.

Você usa chaves de condição para ajudar a evitar o problema “confused deputy”. Para obter mais informações, consulte the section called “Prevenção contra o ataque do “substituto confuso” em todos os serviços”.

Use as seguintes chaves de condição na política para validar as entidades principais que acessam sua chave do KMS. Quando um usuário importa o modelo, AWS KMS verifica se o ARN da versão do modelo de origem corresponde à condição. Se você não incluir uma condição na política, as entidades principais especificadas poderão usar sua chave do KMS para descriptografar qualquer versão do modelo:

- aws: SourceArn — Use essa chave de condição com `kms:GenerateDataKey` as `kms:Decrypt` ações e.
- kms: EncryptionContext — Use essa chave de condição com as `kms:CreateGrant` ações `kms:GenerateDataKeykms:Decrypt`, e.

No exemplo a seguir, a política Conta da AWS `444455556666` autoriza o uso da versão 1 do modelo de classificador especificado de propriedade da. Conta da AWS `111122223333`

Example Política de chaves do KMS para acessar uma versão específica do modelo de classificador

```
{
  "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"
        }
      }
    }
  ]
}
```

O exemplo de política a seguir autoriza o usuário ExampleUser ExampleRolede Conta da AWS 444455556666 e Conta da AWS 123456789012 para acessar essa chave KMS por meio do serviço Amazon Comprehend.

Example Política de chaves do KMS para permitir o acesso ao serviço Amazon Comprehend (alternativa 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:*"
          }
      }
    }
  ]
```

```
}
```

O exemplo de política a seguir Conta da AWS 444455556666 autoriza o acesso a essa chave KMS por meio do serviço Amazon Comprehend, usando uma sintaxe alternativa ao exemplo anterior.

Example Política de chaves do KMS para permitir o acesso ao serviço Amazon Comprehend (alternativa 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:*"
          }
      }
    }
  ]
}
```

Para obter mais informações, consulte Políticas de chaves no AWS KMS no Guia do desenvolvedor do AWS Key Management Service .

Políticas baseadas em recursos para modelos personalizados

Antes que um usuário do Amazon Comprehend em Conta da AWS outro possa importar um modelo personalizado da AWS sua conta, você deve autorizá-lo a fazer isso. Para autorizá-los, adicione uma política baseada em recursos à versão do modelo que deseja compartilhar. Uma política baseada em recursos é uma política do IAM que você anexa a um recurso na AWS.

Quando você anexa uma política de recursos a uma versão de modelo personalizado, a política autoriza usuários, grupos ou funções a executar a ação `comprehend:ImportModel` na versão do modelo.

Example Política baseada em recursos para uma versão de modelo personalizado

Este exemplo especifica as entidades autorizadas no atributo `Principal`. O recurso "*" se refere à versão específica do modelo à qual você anexa a política.

```
{
  "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"
         ]
      }
    }
  ]
}
```

Para políticas que você anexa a modelos personalizados, `comprehend:ImportModel` é a única ação que o Amazon Comprehend suporta.

Para obter mais informações sobre as políticas de acesso baseadas em recursos, consulte Políticas baseadas em identidade no Guia do usuário do IAM.

Etapa 1: incluir uma política baseada em recursos em um modelo personalizado

Você pode adicionar uma política baseada em recursos usando a AWS Management Console, AWS CLI, ou a API Amazon Comprehend.

AWS Management Console

Você pode usar o Amazon Comprehend no AWS Management Console.

Para adicionar uma política baseada em recurso

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu de navegação à esquerda, em Personalização, escolha a página que contém seu modelo personalizado:
   a. Se você estiver compartilhando um classificador personalizado de documento, escolha Classificação personalizada.
   b. Se você estiver compartilhando um reconhecedor personalizado de entidade, escolha Reconhecimento personalizado de entidade.
3. Na lista de modelos, escolha o nome do modelo para exibir sua página de detalhes.
4. Em Versões, escolha o nome da versão do modelo que deseja compartilhar.
5. Na página de detalhes da versão, escolha a guia Tags, VPC e política.
6. Na seção Política baseada em recursos, escolha Editar.
7. Na página Editar página de política baseada em recursos, faça o seguinte:
   a. Em Nome da política, insira um nome que ajude a reconhecer a política após criá-la.
   b. Em Autorizar, especifique uma ou mais das seguintes entidades para autorizá-las a importar seu modelo:

Campo	Definição e exemplos
Entidades principais de serviço	Identificadores de entidades principais de serviço para os serviços que podem acessar essa versão do modelo. Por exemplo: comprehend.amazonaws.com
Conta da AWS IDs	Contas da AWS que podem acessar esta versão do modelo. Autoriza todos os usuários que pertencem à conta. Por exemplo: 111122223333, 123456789012
Entidades do IAM	ARNs para usuários ou funções que podem acessar essa versão do modelo. Por exemplo:

Campo	Definição e exemplos
	arn: aws: iam: :111122223333: user/ExampleUser, arn:aws:iam::444455556666:role/ExampleRole

8. Em Compartilhar, você pode copiar o ARN da versão do modelo para ajudar no compartilhamento com a pessoa que importará seu modelo. Quando alguém importa um modelo personalizado de outro Conta da AWS, a versão ARN do modelo é necessária.
9. Escolha Salvar. O Amazon Comprehend cria sua política baseada em recursos e a anexa ao seu modelo.

AWS CLI

Para adicionar uma política baseada em recursos a um modelo personalizado com o AWS CLI, use o PutResourcePolicycomando. O comando usa os seguintes parâmetros:

- `resource-arn`: o ARN do modelo personalizado, incluindo a versão do modelo.
- `resource-policy`: um arquivo JSON que define a política baseada em recursos a ser anexada ao seu modelo personalizado.

  Você também pode fornecer a política como uma string JSON em linha. Para fornecer JSON válido para sua política, coloque os nomes e valores dos atributos entre aspas duplas. Se o corpo do JSON também estiver entre aspas duplas, você fará o escape das aspas duplas que estão dentro da política.
- `policy-revision-id`: o ID de revisão que o Amazon Comprehend atribuiu à política que você está atualizando. Se você estiver criando uma nova política que não tenha uma versão anterior, não use esse parâmetro. O Amazon Comprehend cria o ID de revisão para você.

Example Adicionar uma política baseada em recursos a um modelo personalizado usando o comando **`put-resource-policy`**

Este exemplo define uma política em um arquivo JSON chamado policyFile.json e associa a política a um modelo. O modelo é a versão v2 de um classificador chamado 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
```

O arquivo JSON da política de recursos contém o seguinte conteúdo:

- Ação: a política autoriza o uso de `comprehend:ImportModel` pelas entidades principais nomeadas.
- Recurso: o ARN do modelo personalizado. O recurso "*" se refere à versão do modelo que você especifica no comando `put-resource-policy`.
- Principal — A política autoriza o usuário de Conta da AWS 444455556666 e todos os usuários `jane` de 123456789012. Conta da 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

Para adicionar uma política baseada em recursos a um modelo personalizado usando a API Amazon Comprehend, use a operação da API. PutResourcePolicy

Você também pode adicionar uma política a um modelo personalizado na solicitação de API que cria o modelo. Para fazer isso, forneça a política JSON para o ModelPolicy parâmetro ao enviar uma CreateEntityRecognizersolicitação CreateDocumentClassifieror.

Etapa 2: fornecer os detalhes que outras pessoas precisam importar

Agora que você adicionou a política baseada em recursos ao seu modelo personalizado, você autorizou outros usuários do Amazon Comprehend a importar seu modelo para as Contas da AWS deles. No entanto, você deve fornecer os seguintes detalhes antes que eles possam importar:

- O nome do recurso da Amazon (ARN) da versão do modelo.
- A Região da AWS que contém o modelo. Qualquer pessoa que importe seu modelo deverá usar a mesma Região da AWS .
- Se o modelo é criptografado e, em caso afirmativo, o tipo de AWS KMS chave que você usa: Chave pertencente à AWS ou chave gerenciada pelo cliente.
- Se seu modelo for criptografado com uma chave gerenciada pelo cliente, será necessário fornecer o ARN da chave do KMS. Qualquer pessoa que importe seu modelo deverá incluir o ARN em um perfil de serviço do IAM em sua Conta da AWS. Esse perfil autoriza o Amazon Comprehend a usar a chave do KMS para descriptografar o modelo durante a importação.

Para obter mais informações sobre como outros usuários importam seu modelo, consulte Importando um modelo personalizado de outro Conta da AWS.

Importando um modelo personalizado de outro Conta da AWS

No Amazon Comprehend, você pode importar um modelo personalizado que está em outro. Conta da AWS Ao importar um modelo, você cria um novo modelo personalizado na sua conta. Seu novo modelo personalizado é uma cópia totalmente treinada do modelo que você importou.

Tópicos

- Antes de começar
- Como importar um modelo personalizado

Antes de começar

Antes de importar um modelo personalizado de outro Conta da AWS, certifique-se de que a pessoa que compartilhou o modelo com você faça o seguinte:

- Autorize você a fazer a importação. Essa autorização é concedida na política baseada em recurso anexada à versão do modelo. Para obter mais informações, consulte Políticas baseadas em recursos para modelos personalizados.

- Forneça a você as informações a seguir:
  - O nome do recurso da Amazon (ARN) da versão do modelo.
  - A Região da AWS que contém o modelo. Você deve usar o mesmo Região da AWS ao importar.
  - Se o modelo está criptografado com uma AWS KMS chave e, se for, o tipo de chave usada.

Se o modelo estiver criptografado, talvez seja necessário executar etapas adicionais, dependendo do tipo de chave do KMS usada:

- Chave pertencente à AWS: esse tipo de chave do KMS pertence e é gerenciado pela AWS. Se o modelo for criptografado com um Chave pertencente à AWS, nenhuma etapa adicional será necessária.
- Chave gerenciada pelo cliente — Esse tipo de chave KMS é criada, de propriedade e gerenciada por um AWS cliente em seu Conta da AWS. Se o modelo for criptografado com uma chave gerenciada pelo cliente, a pessoa que compartilhou o modelo deverá:
  - Autorizar você a descriptografar o modelo. Essa autorização é concedida na política de chaves do KMS para a chave gerenciada pelo cliente. Para obter mais informações, consulte AWS KMS declaração de política chave.
  - Forneça o ARN da chave gerenciada pelo cliente. Você usa esse ARN ao criar um perfil de serviço do IAM. Esse perfil autoriza o Amazon Comprehend a usar a chave do KMS para descriptografar o modelo.

Permissões obrigatórias

Antes de importar um modelo personalizado, você ou seu administrador devem autorizar as ações necessárias no AWS Identity and Access Management (IAM). Como usuário do Amazon Comprehend, você deve estar autorizado a importar por meio de uma declaração de política do IAM. Se a criptografia ou a descriptografia forem necessárias durante a importação, o Amazon Comprehend deverá ser autorizado a usar as chaves necessárias. AWS KMS

Declaração de política do IAM

Seu usuário, grupo ou perfil deve ter uma política anexada que permita a ação `ImportModel`, conforme apresentado no exemplo a seguir.

Example Política do IAM para importar um modelo personalizado

```
{
```

```
  "Effect": "Allow",
  "Action": [
    "comprehend:ImportModel"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Para obter mais informações sobre como criar uma política do IAM, consulte Criar políticas do IAM no Guia do usuário do IAM. Para obter informações sobre como anexar a política do IAM, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

Perfil de serviço do IAM para criptografia do AWS KMS

Ao importar um modelo personalizado, você deve autorizar o Amazon Comprehend a AWS KMS usar chaves em qualquer um dos seguintes casos:

- Você está importando um modelo personalizado que é criptografado com uma chave gerenciada pelo cliente. AWS KMS Nesse caso, o Amazon Comprehend precisará acessar a chave do KMS para poder decifrar o modelo durante a importação.
- Você deseja criptografar o novo modelo personalizado criado com a importação e usar uma chave gerenciada pelo cliente. Nesse caso, o Amazon Comprehend precisará acessar sua chave do KMS para poder criptografar o novo modelo.

Para autorizar o Amazon Comprehend a AWS KMS usar essas chaves, você cria uma função de serviço do IAM. Esse tipo de função do IAM permite que um AWS serviço acesse recursos em outros serviços em seu nome. Para obter mais informações sobre funções de serviço, consulte Criação de uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Se você usar o console do Amazon Comprehend para a importação, poderá fazer com que o Amazon Comprehend crie o perfil de serviço para você. Caso contrário, você deverá criar um perfil de serviço no IAM antes da importação.

O perfil de serviço do IAM deve ter uma política de permissões e uma política de confiança, conforme exibido nos exemplos a seguir.

Example política de permissões

A política de permissões a seguir permite as AWS KMS operações que o Amazon Comprehend usa para criptografar e descriptografar modelos personalizados. Ela concede acesso a duas chaves do KMS:

- Uma chave KMS está na Conta da AWS que contém o modelo a ser importado. Ela foi usada para criptografar o modelo, e o Amazon Comprehend a utilizará para descriptografar o modelo durante a importação.
- A outra chave KMS está na Conta da AWS que importa o modelo. O Amazon Comprehend utilizará essa chave para criptografar o novo modelo personalizado criado pela importação.

```
{
  "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 trust policty (política de confiança)

A seguinte política de confiança permite que o Amazon Comprehend assuma o perfil e obtenha suas permissões. Ela permite que a entidade principal do serviço `comprehend.amazonaws.com` realize a operação `sts:AssumeRole`. Para ajudar na prevenção do problema "confused deputy", você restringe o escopo da permissão usando uma ou mais chaves de contexto de condição global. Para `aws:SourceAccount`, especifique o ID da conta do usuário que está importando o modelo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "444455556666"
        }
      }
    }
  ]
}
```

Como importar um modelo personalizado

Você pode importar um modelo personalizado usando a API AWS Management Console AWS CLI, ou Amazon Comprehend.

AWS Management Console

Você pode usar o Amazon Comprehend no AWS Management Console.

Para importar um modelo personalizado

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu de navegação à esquerda, em Personalização, escolha a página do tipo de modelo que você está importando:

   a. Se você estiver importando um classificador personalizado de documentos, escolha Classificação personalizada.

   b. Se você estiver importando um reconhecedor personalizado de entidade, escolha Reconhecimento personalizado de entidade.
3. Escolha Importar versão.
4. Na página Importar a versão do modelo, digite os seguintes detalhes:

   - ARN da versão do modelo: o ARN da versão do modelo a ser importado.
   - Nome do modelo: um nome personalizado para o novo modelo criado pela importação.
   - Nome da versão: um nome personalizado para a nova versão do modelo criada pela importação.
5. Em Criptografia de modelo, escolha o tipo de chave do KMS a ser usada para criptografar o novo modelo personalizado que você cria com a importação:

   - Use AWS uma chave própria — O Amazon Comprehend criptografa seu modelo usando uma chave AWS Key Management Service in AWS KMS() que é criada, gerenciada e usada em seu nome por. AWS
   - Escolha uma AWS KMS chave diferente (avançada) — O Amazon Comprehend criptografa seu modelo usando uma chave gerenciada pelo cliente que você gerencia. AWS KMS

     Se você escolher essa opção, selecione uma chave KMS que esteja na sua Conta da AWS ou crie uma nova escolhendo Criar uma AWS KMS chave.
6. Na seção Acesso ao serviço, conceda acesso ao Amazon Comprehend para todas as chaves do AWS KMS necessárias para:

   - Descriptografar o modelo personalizado que você importa.
   - Criptografar o novo modelo personalizado que você cria com a importação.

Você concede acesso com um perfil de serviço do IAM que permite que o Amazon Comprehend use as chaves do KMS.

Em Perfil de serviço, faça um dos seguintes procedimentos:

- Se tiver um perfil de serviço que deseja usar, escolha Utilizar um perfil do IAM existente. Em seguida, selecione-o em Nome do perfil.
- Se quiser que o Amazon Comprehend crie um perfil para você, escolha Criar um perfil do IAM.

7. Se tiver optado pela criação de um perfil pelo Amazon Comprehend, faça o seguinte:

   a. Em Nome do perfil, insira um sufixo de nome do perfil que ajude você a reconhecer o perfil posteriormente.

   b. Em ARN da chave do KMS de origem, insira o ARN da chave do KMS que é usada para criptografar o modelo que você está importando. O Amazon Comprehend usa essa chave para descriptografar o modelo durante a importação.

8. (Opcional) Na seção Tags, você pode adicionar tags ao novo modelo personalizado criado na importação. Para obter mais informações sobre tags em modelos personalizados, consulte Marcar um novo recurso.
9. Selecione a opção Confirmar.

AWS CLI

Você pode usar o Amazon Comprehend executando comandos com o AWS CLI.

Example Comando importar-modelo

Para importar um modelo personalizado, use o comando `import-model`:

```
$ 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
```

Este exemplo usa os seguintes parâmetros:

- `source-model`: o ARN do modelo personalizado a ser importado.
- `model-name`: um nome personalizado para o novo modelo criado pela importação.
- `version-name`: um nome personalizado para a nova versão de modelo criada pela importação.
- `data-access-role-arn`— O ARN da função de serviço do IAM que permite ao Amazon Comprehend usar as chaves AWS KMS necessárias para criptografar ou descriptografar o modelo personalizado.
- `model-kms-key-id`: o ARN ou ID da chave do KMS que o Amazon Comprehend usa para criptografar o modelo personalizado que você cria com essa importação. Essa chave deve estar AWS KMS no seu Conta da AWS.

API Amazon Comprehend

Para importar um modelo personalizado usando a API Amazon Comprehend, use ImportModela ação de API.

Flywheels

Um flywheel do Amazon Comprehend simplifica o processo de aprimoramento de um modelo personalizado ao longo do tempo. É possível usar um flywheel para orquestrar as tarefas associadas ao treinamento e à avaliação de novas versões de modelos personalizados. Flywheels são compatíveis com modelos personalizados de texto sem formatação para classificação personalizada e reconhecimento de entidade personalizada.

Tópicos

- Visão geral do flywheel
- Data lakes dos flywheels
- Políticas do IAM e permissões
- Configurar flywheels usando o console
- Configurar flywheels usando a API
- Configurar conjuntos de dados
- Iterações do flywheel
- Utilizar flywheels para análise

Visão geral do flywheel

Um flywheel é um recurso do Amazon Comprehend que orquestra o treinamento e a avaliação de novas versões de um modelo personalizado. Você pode criar um flywheel para usar um modelo treinado existente, ou o Amazon Comprehend pode criar e treinar um novo modelo para o flywheel. Use flywheels com modelos personalizados de texto sem formatação para classificação personalizada ou reconhecimento personalizado de entidades.

Você pode configurar e gerenciar flywheels usando o console ou a API do Amazon Comprehend. Você também pode configurar flywheels usando o AWS CloudFormation.

Ao criar um flywheel, o Amazon Comprehend cria um data lake na sua conta. O data lake armazena e gerencia todos os dados do flywheel, como dados de treinamento e dados de teste de todas as versões do modelo.

Você define a versão do modelo ativo como a versão do modelo do flywheel que deseja usar para tarefas de inferência ou endpoints do Amazon Comprehend. Inicialmente, o flywheel contém uma versão do modelo. Com o tempo, ao treinar novas versões, você seleciona a com melhor

desempenho para ser a versão ativa do modelo. Quando um usuário especifica o ARN do flywheel para executar uma tarefa de inferência, o Amazon Comprehend executa a tarefa usando a versão do modelo ativo do flywheel.

Periodicamente, você obtém novos dados rotulados (dados de treinamento ou dados de teste) para o modelo. Você disponibiliza novos dados para o flywheel criando um ou mais conjuntos de dados. Um conjunto de dados contém dados de entrada para treinar ou testar o modelo personalizado associado a um flywheel. O Amazon Comprehend carrega os dados de entrada no data lake do flywheel.

Para incorporar os novos conjuntos de dados em seu modelo personalizado, você cria e executa uma iteração do flywheel. Uma iteração do flywheel é um fluxo de tarefa que usa os novos conjuntos de dados para avaliar a versão ativa do modelo e treinar uma nova versão dele. Com base nas métricas das versões existentes e novas do modelo, você pode decidir se deseja promover a nova versão do modelo como a versão ativa.

Você pode usar a versão do modelo ativo do flywheel para executar análises personalizadas (tarefas em tempo real ou assíncronas). Para usar o modelo do flywheel para análise em tempo real, você deve criar um endpoint para ele.

Não há cobrança adicional pelo uso de flywheels. No entanto, a execução de uma iteração do flywheel implica em cobranças padrão para treinar uma nova versão do modelo e armazenar os dados dele. Para informações detalhadas sobre preços, consulte Valores do Amazon Comprehend.

Tópicos

- Conjuntos de dados do flywheel
- Criação do flywheel
- Estados do flywheel
- Iterações do flywheel

Conjuntos de dados do flywheel

Para adicionar novos dados rotulados a um flywheel, você cria um conjunto de dados. Configure cada conjunto de dados como dados de treinamento ou dados de teste. Associe o conjunto de dados a um flywheel específico e a um modelo personalizado.

Depois de criar um conjunto de dados, o Amazon Comprehend carrega os dados para o data lake do flywheel. Para obter mais informações, consulte Data lakes dos flywheels.

Criação do flywheel

Associe um flywheel criado a um modelo treinado existente ou o flywheel pode criar um novo modelo.

Ao criar um flywheel com um modelo existente, especifique a versão ativa do modelo. O Amazon Comprehend copia os dados de treinamento e os dados de teste do modelo no data lake do flywheel. Certifique-se de que os dados de treinamento e teste do modelo existam no mesmo local do Amazon S3 em que você criou o modelo.

Para criar um flywheel para um novo modelo, forneça um conjunto de dados para dados de treinamento (e um conjunto de dados opcional para dados de teste) ao criar o flywheel. Ao executar o flywheel para criar a sua primeira iteração, o flywheel treina o novo modelo.

Ao treinar um modelo personalizado, especifique uma lista de rótulos personalizados (classificação personalizada) ou entidades personalizadas (reconhecimento de entidade personalizada) para o modelo reconhecer. Anote os seguintes pontos importantes sobre rótulos/entidades personalizados:

- Ao criar um flywheel para um novo modelo, a lista de rótulos/entidades fornecidas durante a criação do flywheel é a sua lista final.
- Ao criar um flywheel a partir de um modelo existente, a lista de rótulos/entidades associadas a esse modelo se torna a lista final do flywheel.
- Se você associar um novo conjunto de dados ao volante, e esse conjunto de dados contiver outros. labels/entities, Amazon Comprehend ignores the new labels/entities
- Você pode revisar a lista de etiquetas/entidades de um volante usando a operação da API. DescribeFlywheel

> **Note**
>
> Para classificação personalizada, o Amazon Comprehend preenche a lista de rótulos depois que o status do flywheel se torna ATIVO. Espere até que o volante esteja ativo antes de chamar a operação da DescribeFlywheel API.

Estados do flywheel

Um flywheel faz a transição entre os seguintes estados:

- CRIANDO: O Amazon Comprehend está criando os recursos do flywheel. É possível executar operações de leitura no flywheel, como `DescribeFlywheel`.
- ATIVO: O flywheel está ativo. Você pode determinar se uma iteração do flywheel está em andamento e visualizar o status da iteração. É possível realizar ações de leitura no flywheel e ações como `DeleteFlywheel` e `UpdateFlywheel`.
- ATUALIZANDO: O Amazon Comprehend está atualizando o flywheel. É possível executar operações de leitura no flywheel.
- EXCLUINDO: O Amazon Comprehend está excluindo o flywheel. É possível executar operações de leitura no flywheel.
- FALHA: a operação de criação do flywheel falhou.

Depois que o Amazon Comprehend exclui um flywheel, você mantém o acesso a todos os dados do modelo no data lake do flywheel. O Amazon Comprehend exclui todos os metadados internos necessários para gerenciar os recursos do flywheel. O Amazon Comprehend também exclui os conjuntos de dados associados a esse flywheel (os dados do modelo são salvos no data lake).

Iterações do flywheel

Ao obter novos dados de treinamento ou teste para um modelo de flywheel, crie um ou mais novos conjuntos de dados para carregar os novos dados no data lake do flywheel.

Em seguida, execute o flywheel para criar uma nova iteração do flywheel. A iteração do flywheel avalia a versão atual do modelo ativo usando os novos dados e armazena os resultados no data lake. O flywheel também cria e treina uma nova versão do modelo.

Se o novo modelo apresentar melhor desempenho do que a versão atual do modelo ativo, você poderá promover a nova versão do modelo para ser a versão ativa. Você pode usar o console ou a operação UpdateFlywheelda API para atualizar a versão ativa do modelo.

Data lakes dos flywheels

Quando você cria um flywheel, o Amazon Comprehend cria um data lake em sua conta para conter todos os dados do flywheel, como os dados de entrada e saída necessários para as versões do modelo.

O Amazon Comprehend cria o data lake no local do Amazon S3 que você especifica ao criar o flywheels. Você pode especificar o local como um bucket do Amazon S3 ou como uma nova pasta em um bucket do Amazon S3.

Estrutura de pastas do data lake

Quando o Amazon Comprehend cria o data lake, ele configura a seguinte estrutura de pastas no local do Amazon S3.

> **Warning**
>
> O Amazon Comprehend gerencia a organização e o conteúdo das pastas do data lake. Sempre use as operações da API do Amazon Comprehend para modificar as pastas do data lake, ou seu flywheel pode não funcionar corretamente.

```
Document Pool
Annotations Pool
Staging
Model Datasets
  (data for each version of the model)
  VersionID-1
    Training
    Test
    ModelStats
  VersionID-2
    Training
    Test
    ModelStats
```

Para visualizar a avaliação de treinamento de uma versão do modelo, execute estas etapas:

1. Abra a pasta chamada Conjunto de dados do modelo no nível raiz do data lake. Essa pasta contém uma subpasta para cada versão do modelo.
2. Abra a pasta da versão do modelo de interesse.
3. Abra a pasta nomeada ModelStatspara ver as estatísticas do modelo.

Gerenciamento do data lake

O Amazon Comprehend realiza as seguintes tarefas para gerenciar o data lake em seu nome:

- Define a estrutura de pastas do data lake e ingere os conjuntos de dados nas pastas apropriadas.
- Gerencia os documentos de entrada (como arquivos de texto e arquivos de anotações) necessários para treinar o modelo.
- Gerencia os dados de saída de treinamento e avaliação associados a cada versão do modelo.
- Gerencia a criptografia dos arquivos armazenados no data lake.

O Amazon Comprehend executa todas as operações de criação e atualização de dados para o data lake. Você retém o acesso total aos dados no data lake. Por exemplo:

- Você tem acesso total ao conteúdo do data lake.
- O data lake permanece disponível depois que você exclui o .flywheel
- Você pode configurar os logs de acesso ao bucket do Amazon S3 que contém o data lake.
- Você pode fornecer chaves de criptografia para os dados. Você as especifica ao criar um flywheel.

Recomendamos seguir estas práticas recomendadas:

- Não adicione suas próprias pastas ou arquivos ao data lake manualmente. Não modifique nem exclua nenhum arquivo no data lake.
- Sempre use as operações de criação e atualização do Amazon Comprehend para adicionar ou modificar dados no data lake. Por exemplo, use `CreateDataset` para fornecer dados de treinamento ou teste e `StartFlywheelIteration` para gerar dados de avaliação para versões do modelo.
- A estrutura do data lake pode evoluir com o tempo. Não crie scripts ou programas downstream que dependam explicitamente da estrutura do data lake.
- Ao fornecer uma localização de data lake para o flywheel, recomendamos criar um prefixo comum para dados relacionados a todos os flywheels ou usar um prefixo diferente para cada flywheel. Não recomendamos usar o caminho completo do data lake de um flywheel como prefixo para outro flywheel.

Políticas do IAM e permissões

Você configura as seguintes políticas e permissões para usar flywheel:

- the section called “Configure as permissões de usuário do IAM” para que os usuários acessem as operações do flywheel.
- (Opcional) the section called “Configurar permissões para AWS KMS chaves” para o data lake.
- the section called “Criar um perfil de acesso a dados” que autoriza o Amazon Comprehend a acessar o data lake.

Configure as permissões de usuário do IAM

Para usar os recursos do flywheel, adicione políticas de permissões apropriadas às suas identidades AWS Identity and Access Management (IAM) (usuários, grupos e funções).

O exemplo a seguir mostra a política de permissões para criar conjuntos de dados, criar, gerenciar e executar o flywheel.

Example Política do IAM para gerenciar flywheels

```
{
  "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": "*"
}
```

Para mais informações sobre como criar políticas do IAM no Amazon Comprehend, consulte Como o Amazon Comprehend funciona com o IAM.

Configurar permissões para AWS KMS chaves

Se você estiver usando AWS KMS chaves para seus dados no data lake, configure as permissões necessárias. Para mais informações, consulte Permissões necessárias para usar a criptografia KMS.

Criar um perfil de acesso a dados

Crie um perfil de acesso a dados no IAM para que o Amazon Comprehend acesse dados do flywheel no data lake. Se usar o console para criar um flywheel, o sistema poderá, opcionalmente, criar um novo perfil para essa finalidade. Para obter mais informações, consulte Permissões baseadas em perfis necessárias para operações assíncronas.

Configurar flywheels usando o console

É possível usar o console do Amazon Comprehend para criar, atualizar e excluir flywheels.

Quando você cria um flywheel, o Amazon Comprehend cria um data lake para armazenar todos os dados de que o flywheel precisa, como dados de treinamento e dados de teste para cada versão do modelo.

Quando você exclui um flywheel, o Amazon Comprehend não exclui o data lake nem o modelo associado ao flywheel.

Revise as informações na seção Criação do flywheel antes de criar um novo flywheel.

Tópicos

- Crie um flywheel
- Atualize um flywheel
- Exclua um flywheel

Crie um flywheel

Quando você cria um flywheel, os campos de configuração necessários dependem se o flywheel é para um modelo personalizado existente ou para um novo modelo.

Para criar um flywheel

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.

2. No menu à esquerda, escolha Flywheels.
3. Na tabela Flywheels, escolha Criar novo flywheel.
4. Em Nome do flywheel, insira um nome para o flywheel.
5. (Opcional) Para criar um flywheel para um modelo existente, configure os campos em Versão ativa do modelo.

    a. Na lista suspensa Modelo, selecione um modelo.
    b. Na lista suspensa Versão, selecione a versão do modelo.
6. (Opcional) Para criar um novo modelo classificador para o flywheel, em Tipo de modelo personalizado, escolha uma Classificação personalizada e configure os parâmetros nas etapas a seguir.

    a. Em Idioma, selecione o idioma do modelo.
    b. Em Modo do classificador, escolha o modo de rótulo único ou o modo de vários rótulos.
    c. Em Rótulos personalizados, insira um ou mais rótulos personalizados para usar no treinamento do modelo. Cada rótulo deve corresponder a uma das classes em seus dados de treinamento de entrada.
7. (Opcional) Para criar um novo modelo de reconhecimento de entidade para o flywheel, em Tipo de modelo personalizado, escolha um Reconhecimento personalizado de entidade e configure os parâmetros nas etapas a seguir.

    a. Em Idioma, selecione o idioma do modelo.
    b. Em Tipo de entidade personalizada, insira até 25 entidades personalizadas para usar no treinamento do modelo. Cada rótulo deve corresponder a um dos tipos de entidade em seus dados de treinamento de entrada.

        Para criar mais de um rótulo, execute as etapas a seguir várias vezes.

        i. Insira um rótulo personalizado. O rótulo deve estar em maiúsculas. Use um sublinhado como separador entre as palavras no rótulo.
        ii. Escolha Adicionar tipo.

        Para remover um dos rótulos que você adicionou, escolha X à direita do nome do rótulo.
8. Configure suas opções para criptografia de volume, criptografia de modelo e criptografia de data lake. Para cada uma delas, escolha se deseja usar uma AWS chave KMS própria ou uma chave que você tenha permissão para usar.

- Se você estiver usando uma AWS chave KMS própria, não há parâmetros adicionais.
- Se você estiver usando outra chave existente, para ARN da chave do KMS, insira o ARN para o ID da chave.
- Se você quiser criar uma nova chave, escolha Criar uma chave AWS KMS.

Para mais informações sobre como criar e usar chaves KMS e a criptografia associada, consulte AWS Key Management Service.

a. Configure a chave de Criptografia de volume. O Amazon Comprehend usa essa chave para criptografar os dados no volume de armazenamento enquanto seu trabalho está sendo processado. Escolha se deseja usar uma chave KMS própria ou AWS uma chave que você tenha permissão para usar.

b. Configure a chave de Criptografia do modelo. O Amazon Comprehend usa essa chave para criptografar os dados do modelo para essa versão do modelo.

9. Configure a Localização do data lake. Para obter mais informações, consulte Gerenciamento do data lake.
10. (Opcional) Configure a chave de Criptografia do Data Lake. O Amazon Comprehend usa essa chave para criptografar todos os arquivos no data lake.
11. (Opcional) Defina as configurações da VPC. Insira o ID da VPC VPC ou escolha o ID na lista suspensa.

    1. Escolha a sub-rede em Sub-redes. Depois de selecionar a primeira sub-rede, é possível escolher outras adicionais.
    2. Em Grupo(s) de segurança, escolha o grupo de segurança a ser usado se tiver especificado um. Depois de selecionar o primeiro grupo de segurança, é possível escolher outros.

12. Configure as permissões de Acesso ao serviço.

    1. Se você selecionar Utilizar um perfil do IAM existente, selecione o nome do perfil na lista suspensa.
    2. Se você selecionar Criar um perfil do IAM, o Amazon Comprehend criará um novo perfil. O console exibe as permissões que o Amazon Comprehend configura para o perfil. Em Nome do perfil, insira um nome descritivo para o perfil.

13. (Opcional) Defina as configurações em Tags. Para adicionar uma tag, insira um par chave-valor em Tags. Escolha Adicionar Tag. Para remover esse par antes de criar o flywheel, escolha Remover tag. Para obter mais informações, consulte Marcando seus Recursos .
14. Escolha Criar.

Atualize um flywheel

Você só pode configurar o nome do flywheel, a localização do data lake, o tipo de modelo e a configuração do modelo ao criar o flywheel.

Ao atualizar um flywheel, você pode especificar um modelo diferente se o tipo de modelo e as opções de configuração forem iguais ao modelo atual. Você pode configurar uma nova versão ativa do modelo. Você também pode atualizar detalhes de criptografia, permissões de acesso ao serviço e configurações da VPC.

Para atualizar um flywheel

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.
2. No menu à esquerda, escolha Flywheels.
3. Na tabela Flywheels, escolha um flywheel para atualizar.
4. Em Versão ativa do modelo, escolha um modelo na lista suspensa Modelo e escolha uma versão para o modelo.

   O formulário preenche o tipo de modelo e a configuração do modelo.
5. (Opcional) Defina as configurações em Criptografia do volume e Criptografia do modelo.
6. (Opcional) Defina as configurações de Criptografia do data lake.
7. Configure as permissões de Acesso ao serviço.
8. (Opcional) Defina as configurações da VPC.
9. (Opcional) Defina as configurações em Tags.
10. Escolha Salvar .

Exclua um flywheel

Para excluir um flywheel

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.

2. No menu à esquerda, escolha Flywheels.
3. Na tabela Flywheels, escolha o flywheel a ser excluído.
4. Escolha Excluir.

Configurar flywheels usando a API

Você pode usar a API do Amazon Comprehend para criar, atualizar e excluir flywheels.

Quando você cria um flywheel, o Amazon Comprehend cria um data lake para armazenar todos os dados de que o flywheel precisa, como dados de treinamento e dados de teste para cada versão do modelo.

Quando você exclui um flywheel, o Amazon Comprehend não exclui o data lake nem o modelo associado ao flywheel.

A operação de exclusão do flywheel falhará se o flywheel estiver executando uma iteração ou criando um conjunto de dados.

Revise as informações na seção Criação do flywheel antes de criar um novo flywheel.

Crie um flywheel para um modelo existente

Use a CreateFlywheeloperação para criar um volante para um modelo existente.

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"   \
```

Se a operação tiver êxito, a resposta incluirá o ARN do flywheel.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Crie um flywheel para um novo modelo

Use a CreateFlywheeloperação para criar um volante para um novo modelo de classificação personalizado.

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
```

O arquivo taskConfig.json contém o conteúdo a seguir.

```
{
    "LanguageCode": "en",
    "DocumentClassificationConfig": {
        "Mode": "MULTI_LABEL",
        "Labels": ["optimism", "anger"]
    }
}
```

O corpo da resposta da API inclui o conteúdo a seguir.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Descreva um flywheel.

Use a operação Amazon DescribeFlywheelComprehend para recuperar informações configuradas sobre um volante.

```
aws comprehend describe-flywheel \
    --flywheel-arn  "flywheelArn"
```

O corpo da resposta da API inclui o conteúdo a seguir.

```
{
  "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
  }
}
```

Atualize um flywheel

Use a UpdateFlywheeloperação para atualizar os valores de configuração modificáveis do volante.

Alguns campos de configuração são estruturas JSON com subcampos. Para atualizar um ou mais subcampos, forneça valores para todos os subcampos (o Amazon Comprehend define o valor como nulo para qualquer subcampo ausente na solicitação).

Se você omitir um parâmetro de nível superior na solicitação `UpdateFlywheel`, o Amazon Comprehend não irá alterar os valores do parâmetro ou de nenhum de seus subcampos no flywheel.

Para adicionar ou remover etiquetas no volante, use as operações TagResourcee. UntagResource

É possível promover uma versão do modelo definindo o parâmetro `ActiveModelArn`, conforme mostrado no exemplo a seguir.

```
aws comprehend update-flywheel \
    --region aws-region \
    --flywheel-arn  "flywheelArn" \
    --active-model-arn  "modelArn" \
```

O corpo da resposta da API inclui o conteúdo a seguir.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Exclua um flywheel

Use a operação Amazon DeleteFlywheelComprehend para excluir volantes.

```
aws comprehend delete-flywheel \
    --flywheel-arn  "flywheelArn"
```

Uma resposta de API bem-sucedida contém um de mensagens corpo vazio.

Liste os flywheels

Use a operação Amazon ListFlywheelsComprehend para recuperar uma lista de volantes na região atual.

```
aws comprehend list-flywheel \
    --region aws-region \
    --endpoint-url  "uri"
```

O corpo da resposta da API inclui o conteúdo a seguir.

```
{
    "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
        }
    ]
}
```

Configurar conjuntos de dados

Para adicionar dados de treinamento ou teste rotulados a um flywheel, use o console ou a API do Amazon Comprehend para criar um conjunto de dados.

Configure cada conjunto de dados como dados de treinamento ou dados de teste. Associe o conjunto de dados a um flywheel específico e a um modelo personalizado. Ao criar um conjunto de dados, o Amazon Comprehend carrega os dados para o data lake do flywheel. Para obter detalhes sobre formatos de arquivo para os dados de treinamento, consulte Preparar dados de treinamento do classificador ou Preparando dados de treinamento do reconhecedor de entidades.

Quando você exclui o flywheel, o Amazon Comprehend exclui os conjuntos de dados. Os dados enviados permanecem disponíveis no data lake.

Criar um conjunto de dados (console)

Criar um conjunto de dados

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.
2. No menu à esquerda, escolha Flywheels e escolha o flywheel em que você deseja adicionar os dados.
3. Escolha a guia Conjuntos de dados.
4. Na tabela Conjuntos de dados de treinamento ou Conjuntos de dados de teste, escolha Criar conjunto de dados.
5. Em Detalhes do conjunto de dados, insira um nome para o conjunto de dados e uma descrição opcional.
6. Em Especificações de dados, escolha os campos de configuração Formato de dados e Tipo de conjunto de dados.
7. (Opcional) Em Formato de entrada, escolha o formato dos documentos de entrada.
8. Em Local da anotação no S3, insira a localização do arquivo de anotações do Amazon S3.
9. Em Local dos dados de treinamento no S3, insira a localização dos arquivos de documentos do Amazon S3.
10. Escolha Criar.

Criar um conjunto de dados (API)

Você pode usar a CreateDatasetoperação para criar um conjunto de dados.

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
}
```

O arquivo `inputConfig.json` contém o conteúdo a seguir.

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://my-comprehend-datasets/multilabel_train.csv"
    }
}
```

Para adicionar ou remover tags no conjunto de dados, use as UntagResourceoperações TagResourcee.

Descrever um conjunto de dados

Use a operação Amazon DescribeDatasetComprehend para recuperar informações configuradas sobre um volante.

```
aws comprehend describe-dataset \
    --dataset-arn  "datasetARN"
```

A resposta tem o conteúdo a seguir.

```
{
   "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
  }
}
```

Iterações do flywheel

Use as iterações do flywheel para facilitar a criação e o gerenciamento de novas versões do modelo.

Tópicos

- Fluxo de trabalho da iteração
- Gerenciamento de iterações (console)
- Gerenciamento de iterações (API)

Fluxo de trabalho da iteração

Um flywheel começa com uma versão treinada do modelo ou usa um conjunto de dados inicial para treinar uma versão do modelo.

Com o tempo, à medida que obtém novos dados rotulados, você treina novas versões do modelo para melhorar o desempenho do seu modelo de flywheel. Quando você executa o flywheel, ele cria uma nova iteração que treina e avalia uma nova versão do modelo. Você pode promover a nova versão do modelo se o seu desempenho for superior ao da versão atual do modelo ativo.

Este fluxo de trabalho de iteração do flywheel inclui as seguintes etapas:

1. Criação de conjuntos de dados para os novos dados rotulados.
2. Execução do flywheel para criar uma nova iteração. A iteração segue estas etapas para treinar e avaliar uma nova versão do modelo:
   a. Avaliação da versão ativa do modelo usando os novos dados.
   b. Treinamento de uma nova versão do modelo usando os novos dados.

c. Armazenamento dos resultados da avaliação e do treinamento no data lake.

d. Retorno dos F1 scores para ambos os modelos.

3. Depois que a iteração for concluída, você pode comparar os F1 scores do modelo ativo existente e do novo modelo.
4. Se a versão do novo modelo tiver desempenho superior, você pode promovê-lo para ser a versão de modelo ativo. Use o console ou a API para promover a nova versão do modelo.

Gerenciamento de iterações (console)

Use o console para iniciar uma nova iteração e consultar o status de uma iteração em andamento. Você também pode visualizar os resultados das iterações concluídas.

Iniciar uma iteração do flywheel (console)

Antes de iniciar uma nova iteração, crie um ou mais novos conjuntos de dados de treinamento ou teste. Consulte Configurar conjuntos de dados

Iniciar uma iteração do flywheel (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.
2. No menu à esquerda, escolha Flywheels.
3. Na tabela Flywheels, escolha um flywheel.
4. Selecione Executar flywheel.

Analise os resultados da iteração (console)

Depois de executar a iteração do flywheel, o console exibe os resultados na tabela de Iterações dos flywheels.

Promova a nova versão do modelo (Console)

Na página de detalhes do modelo no console, promova uma nova versão do modelo para ser a versão ativa.

Promova uma versão do modelo do flywheel para a versão do modelo ativo (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend.

2. No menu à esquerda, escolha Flywheels.
3. Na tabela Flywheels, escolha um flywheel.
4. Na tabela da Página de detalhes do flywheel, escolha a versão a ser promovida na tabela de Iterações dos flywheels.
5. Escolha Tornar modelo ativo.

Gerenciamento de iterações (API)

Você pode usar a API do Amazon Comprehend para iniciar uma nova iteração e consultar o status de uma iteração em andamento. Você também pode visualizar os resultados das iterações concluídas.

Inicie a iteração do flywheel (API)

Use a operação Amazon StartFlywheelIterationComprehend para iniciar uma iteração do flywheel.

```
aws comprehend start-flywheel-iteration \
    --flywheel-arn  "flywheelArn"
```

A resposta tem o conteúdo a seguir.

```
{
  "FlywheelIterationArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name"
}
```

Promova a nova versão do modelo (API)

Use a UpdateFlywheeloperação para promover uma versão do modelo para ser a versão ativa do modelo.

Envie a solicitação `UpdateFlywheel` com o parâmetro `ActiveModelArn` definido para o ARN da nova versão ativa do modelo.

```
aws comprehend update-flywheel \
    --active-model-arn  "modelArn" \
```

A resposta tem o conteúdo a seguir.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Descrever os resultados da iteração do flywheel (API)

A operação Amazon DescribeFlywheelIterationComprehend retorna informações sobre uma iteração depois que ela é executada até a conclusão.

```
aws comprehend describe-flywheel-iteration \
 --flywheel-arn "flywheelArn" \
 --flywheel-iteration-id  "flywheelIterationId" \
 --region aws-region
```

A resposta tem o conteúdo a seguir.

```
{
    "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>
        }
    }
}
```

Obtenha o histórico de iterações (API)

Use a ListFlywheelIterationHistoryoperação para obter informações sobre o histórico de iterações.

```
aws comprehend list-flywheel-iteration-history \
 --flywheel-arn "flywheelArn"
```

A resposta tem o conteúdo a seguir.

```
{
    "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
            }
        }
    ]
}
```

Utilizar flywheels para análise

Você pode usar a versão do modelo ativo do flywheels para executar análises para classificação personalizada ou reconhecimento de entidades. A versão ativa do modelo é configurável. Você pode

usar o console ou a operação da UpdateFlywheelAPI para definir uma nova versão do modelo como a versão ativa do modelo.

Para usar o flywheel, especifique o ARN do flywheel em vez de um ARN de modelo personalizado ao configurar a tarefa de análise. O Amazon Comprehend executa a análise usando a versão do modelo ativo do flywheel.

Análise em tempo real

Você usa um endpoint para executar a análise em tempo real. Ao criar ou atualizar um endpoint, você pode configurá-lo com o ARN do flywheel em vez de um ARN do modelo. Ao executar a análise em tempo real, selecione o endpoint associado ao flywheel. O Amazon Comprehend executa a análise usando a versão do modelo ativo do flywheel.

Quando você usa UpdateFlywheelpara definir uma nova versão do modelo ativo para o volante, o endpoint é atualizado automaticamente para começar a usar a nova versão do modelo ativo. Se você não quiser que o endpoint seja atualizado automaticamente, configure o endpoint (usando UpdateEndpoint) para usar diretamente o ARN da versão do modelo. O endpoint continua usando essa versão do modelo se a versão do modelo ativo do flywheel mudar.

Para classificação personalizada, use a operação de ClassifyDocumentAPI. Para reconhecimento personalizado de entidades, use a solicitação de DetectEntitiesAPI. Forneça o endpoint do flywheel no parâmetro EndpointArn.

Você também pode usar o console para executar análises em tempo real para classificações personalizadas ou reconhecimentos personalizados de entidades.

Trabalhos assíncronos

Para classificação personalizada, use a solicitação de StartDocumentClassificationJobAPI para iniciar um trabalho assíncrono. Forneça o parâmetro FlywheelArn em vez do DocumentClassifierArn.

Para reconhecimento personalizado de entidades, use a solicitação de StartEntitiesDetectionJobAPI. Forneça o parâmetro FlywheelArn em vez do EntityRecognizerArn.

Você pode usar o console para executar trabalhos de análise assíncrona para classificações personalizadas ou reconhecimentos personalizados de entidades. Ao criar o trabalho, insira o ARN do flywheel no campo Modelo do reconhecedor ou Modelo do classificador.

Gerenciar endpoints do Amazon Comprehend

No Amazon Comprehend, os endpoints disponibilizam seus modelos personalizados para classificação em tempo real ou detecção de entidades. Após criar um endpoint, você pode alterá-lo à medida que suas necessidades de negócios evoluem. Por exemplo, você pode monitorar a utilização do endpoint e aplicar o ajuste de escala automático para definir automaticamente o provisionamento de endpoints de acordo com suas necessidades de capacidade. Você pode gerenciar todos os seus endpoints a partir de uma única visualização e, quando não precisar mais de um endpoint, poderá excluí-lo para economizar custos.

Antes de gerenciar um endpoint, você deve criar um. Para obter mais informações, consulte os procedimentos a seguir:

- Criar um endpoint para classificação personalizada
- Criação de um endpoint para detecção de entidades personalizadas

Tópicos

- Visão geral dos endpoints do Amazon Comprehend
- Utilizar endpoints do Amazon Comprehend
- Monitoramento de endpoints do Amazon Comprehend
- Atualizando os endpoints do Amazon Comprehend
- Usando Trusted Advisor com o Amazon Comprehend
- Excluindo endpoints do Amazon Comprehend
- Ajuste de escala automático com endpoints

Visão geral dos endpoints do Amazon Comprehend

A página de endpoints do console do Amazon Comprehend fornece uma visão global dos seus endpoints. Na página de visão geral dos endpoints, você pode visualizar todos os seus endpoints em um só lugar para entender o uso do endpoint em comparação com o uso real dos recursos. No canto superior direito da página de endpoints, você pode especificar quais endpoints deseja visualizar: todos eles, endpoints de classificadores personalizados ou endpoints personalizados de sua entidade.

É possível criar, atualizar, monitorar e excluir endpoints nesta página. Na seção de visão geral dos endpoints, você pode ver uma lista de seus endpoints, quais modelos personalizados os endpoints estão hospedando, a hora de criação, a throughput provisionada e o status do endpoint. Quando você seleciona um endpoint específico na tabela de visão geral do endpoint, os detalhes do endpoint são exibidos.

Além disso, se você for cliente do AWS Business Support ou do AWS Enterprise Support, você terá acesso às verificações do Trusted Advisor específicas para seus endpoints. Para saber mais, consulte Usando Trusted Advisor com o Amazon Comprehend. Para obter uma lista completa de verificações e descrições, consulte as Práticas recomendadas do Trusted Advisor.

Para obter mais informações sobre o gerenciamento de seus endpoints, consulte os seguintes tópicos.

- Utilizar endpoints do Amazon Comprehend
- Monitoramento de endpoints do Amazon Comprehend
- Atualizando os endpoints do Amazon Comprehend
- Usando Trusted Advisor com o Amazon Comprehend
- Excluindo endpoints do Amazon Comprehend

> **Important**
>
> O custo da classificação personalizada em tempo real é baseado tanto no throughput, quanto no período em que o endpoint está ativo. Se você não estiver mais usando o endpoint ou não o estiver usando por um longo período, configure uma políticas de ajuste de escala automático para reduzir seus custos. Ou, se você não estiver mais usando um endpoint, você poderá excluí-lo para evitar custos adicionais. Para obter mais informações, consulte Ajuste de escala automático com endpoints.

Utilizar endpoints do Amazon Comprehend

Você cria um endpoint para executar análises em tempo real usando um modelo de classificação personalizado. Um endpoint inclui recursos gerenciados que disponibilizam seu modelo personalizado para inferência em tempo real.

O Amazon Comprehend atribui o throughput a um endpoint usando unidades de inferência (IU). Um IU representa um throughput de 100 caracteres por segundo. Você pode provisionar o endpoint com até 10 unidades de inferência. Você pode escalar ou diminuir o throughput do endpoint atualizando o endpoint.

Se seus documentos de entrada incluírem documentos semiestruturados ou arquivos de imagem, o throughput de 100 caracteres por segundo é para os caracteres extraídos do arquivo de entrada. O número IUs que você provisiona para um endpoint depende da densidade de caracteres dos documentos de entrada.

As respostas ClassifyDocumente DetectEntitiesda API incluem a contagem de caracteres para cada página de entrada. Você pode usar essas informações para estimar o número de unidades de inferência a serem provisionadas para alcançar o throughput desejado.

Depois de concluir sua análise em tempo real, exclua o endpoint porque a cobrança por ele continuará enquanto ele estiver ativo. Você pode criar outro endpoint quando estiver pronto para executar mais análises em tempo real.

Para obter mais informações sobre os custos dos endpoint, consulte Preços do Amazon Comprehend.

Depois de criar um endpoint, você pode monitorá-lo com a Amazon CloudWatch, atualizá-lo para alterar suas unidades de inferência ou excluí-lo quando não for mais necessário. Para obter mais informações, consulte Monitoramento de endpoints do Amazon Comprehend.

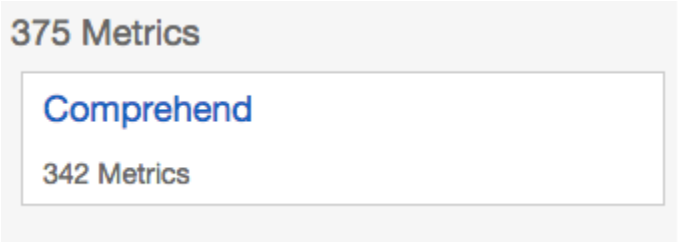
Monitoramento de endpoints do Amazon Comprehend

Você pode ajustar a taxa de transferência do seu endpoint aumentando ou diminuindo o número de unidades de inferência (). IUs Para obter mais informações sobre a atualização do seu endpoint, consulte the section called “Atualizando endpoints”.

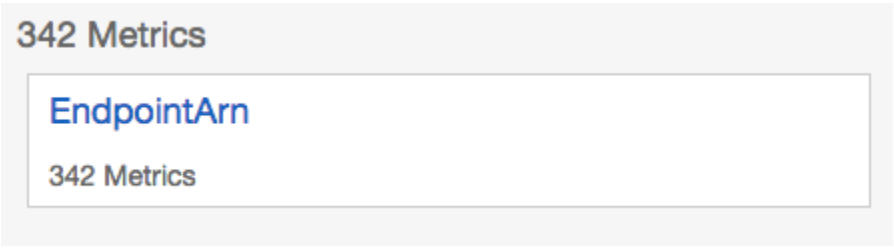
Você pode determinar a melhor forma de ajustar a taxa de transferência do seu endpoint monitorando seu uso com o console da Amazon CloudWatch .

Monitore o uso do seu endpoint com CloudWatch

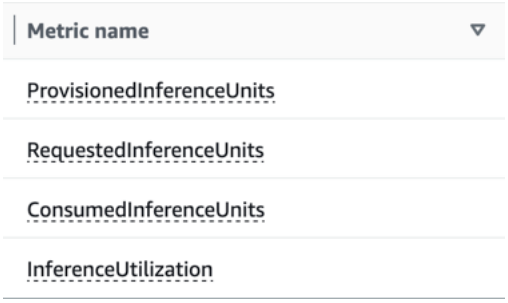
1. Faça login no AWS Management Console e abra o CloudWatch console.
2. À esquerda, escolha Métricas e selecione Todas as métricas.
3. Em Todas as métricas, escolha Comprehend.



4. O CloudWatch console exibe as dimensões das métricas do Comprehend. Escolha a EndpointArndimensão.



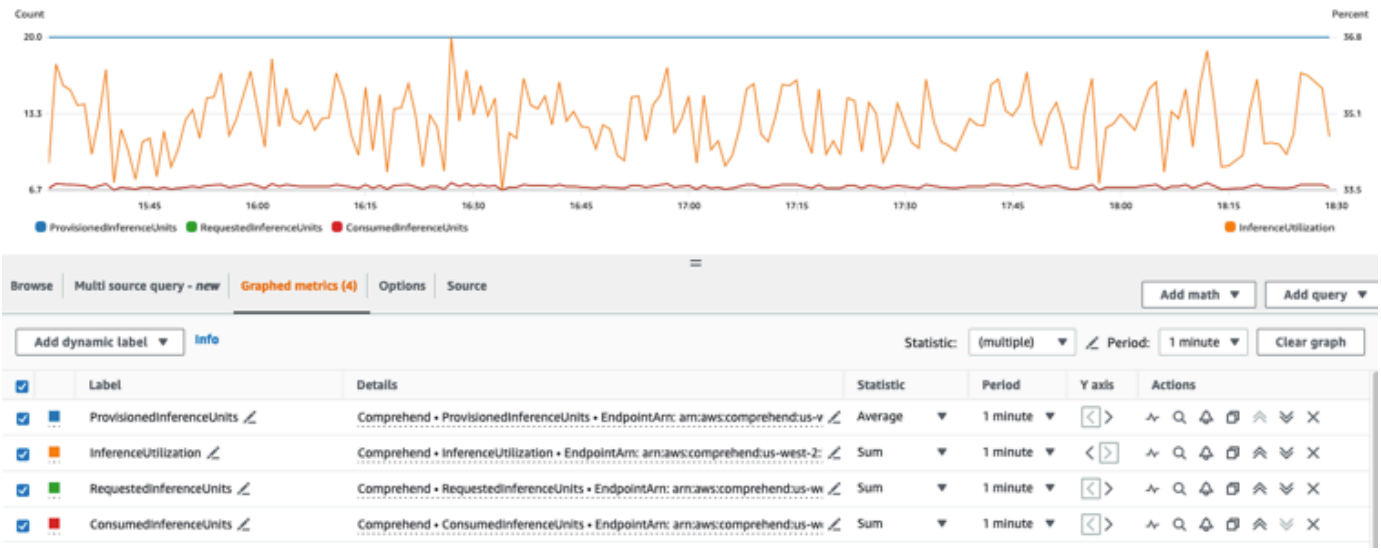
   O console exibe ProvisionedInferenceUnits,, RequestedInferenceUnitsConsumedInferenceUnits, e InferenceUtilizationpara cada um dos seus endpoints.



   Selecione as quatro métricas e navegue até a guia Métricas representadas graficamente.

5. Defina as colunas de estatística para RequestedInferenceUnitse ConsumedInferenceUnitspara Soma.
6. Defina a coluna Estatística InferenceUtilizationpara Soma.
7. Defina a coluna Estatística como Média. ProvisionedInferenceUnits
8. Altere a coluna Período de todas as métricas para 1 minuto.
9. Selecione InferenceUtilizatione selecione a seta para movê-la para um eixo Y separado.

   Seu gráfico está pronto para análise.



Com base nas CloudWatch métricas, você também pode configurar o escalonamento automático para ajustar automaticamente a taxa de transferência do seu endpoint. Para obter mais informações sobre como usar o ajuste de escala automático com os seus endpoints, consulte Ajuste de escala automático com endpoints.

- ProvisionedInferenceUnits- Essa métrica representa o número médio de provisionados IUs no momento em que a solicitação foi feita.
- RequestedInferenceUnits- Isso se baseia no uso de cada solicitação enviada ao serviço que foi enviada para ser processada. Isso pode ser útil para comparar a solicitação enviada para ser processada com a que foi realmente processada sem obter throttling ()ConsumedInferenceUnits. O valor dessa métrica é calculado tomando o número de caracteres enviados para serem processados e dividindo-o pelo número de caracteres que podem ser processados em um minuto por 1 UI.
- ConsumedInferenceUnits- Isso se baseia no uso de cada solicitação enviada ao serviço que foi processada com sucesso (não limitada). Isso pode ser útil quando você compara o que está consumindo com o provisionado IUs. O valor dessa métrica é calculado tomando-se o número de caracteres processados e dividindo-o pelo número de caracteres que podem ser processados em um minuto por 1 UI.
- InferenceUtilization- Isso é emitido por solicitação. Esse valor é calculado tomando o consumo IUs definido em ConsumedInferenceUnits, dividindo-o por ProvisionedInferenceUnitse convertendo em uma porcentagem de 100.

> **Note**
>
> Todas as métricas são emitidas somente para solicitações bem-sucedidas. A métrica não aparecerá se for proveniente de uma solicitação que foi limitada ou falhou devido a um erro interno do servidor ou do cliente.

Atualizando os endpoints do Amazon Comprehend

Frequentemente, o nível de throughput que você precisa muda após a criação de um endpoint ou, então, a primeira estimativa de suas necessidades muda. Quando isso acontece, pode ser necessário atualizar seu endpoint para aumentar ou diminuir o throughput. O throughput é controlada pelo número de unidades de inferência com as quais você provisionou seu endpoint. Cada unidade de inferência representa um throughput de 100 caracteres por segundo para até dois documentos por segundo. Talvez você também queira atualizar a versão do modelo associada ao endpoint. Ao editar um endpoint, você pode escolher uma versão diferente do modelo para o endpoint.

Também pode ser útil adicionar tags ao seu endpoint para ajudar a mantê-los organizados. Isso também pode ser feito ao atualizar o endpoint. Para obter mais informações sobre endpoints, consulte Marcando seus Recursos .

Para atualizar um endpoint (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Endpoints.
3. Na lista Classificadores, escolha o nome do modelo personalizado do qual você deseja atualizar o endpoint e siga o link. A página de detalhes do modelo é exibida.
4. Na página de detalhes do modelo, selecione os detalhes da versão. A lista de endpoints é exibida.
5. Selecione a caixa de seleção do endpoint para seu endpoint. No canto superior direito da tabela de endpoints, selecione o ícone Ações.
6. Selecione Editar. Você pode atualizar as tags provisionadas IUs e editar.
7. Salve as alterações.
8. Para editar o número de unidades de inferência com as quais o endpoint é provisionado, escolha Editar.

9. Insira o número atualizado de unidades de inferência a serem atribuídas ao endpoint. Cada unidade representa um throughput de 100 caracteres por segundo. Você pode atribuir até 10 unidades de inferência por endpoint no máximo.

   > **Note**
   >
   > O custo do uso de um endpoint é baseado na quantidade de tempo da operação e de throughput (com base no número de unidades de inferência). Aumentar o número de unidades de inferência aumentará, portanto, o custo da operação. Para mais informações, consulte Preços do Amazon Comprehend.

10. Escolha Editar endpoint. A página Detalhes do endpoint será exibida.
11. Confirme se o endpoint está sendo atualizado escolhendo o nome do modelo nos rastros de navegação no topo da página. Na página de detalhes do modelo personalizado, navegue até a lista de Endpoints e verifique se ela mostra Atualizando ao lado do endpoint. Quando a atualização for concluída, ela mostrará Pronto.

    O exemplo a seguir demonstra o uso da UpdateEndpointoperação com a AWS CLI.

    O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

    ```
    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
    ```

    Se a ação for executada com êxito, o Amazon Comprehend responderá com uma resposta HTTP 200 com um corpo HTTP vazio.
12. Para editar o modelo personalizado anexado ao seu endpoint, na página de detalhes do modelo personalizado, navegue até a lista de Endpoints.
13. Selecione o endpoint que você deseja alterar e selecione Editar.
14. Na página de configurações do endpoint, em Selecionar modelo do classificador ou Selecionar modelo do reconhecedor dependendo do seu endpoint, você pode pesquisar um modelo no menu suspenso. Selecione o modelo que você quiser.

15. Em Selecionar versão, você pode pesquisar a versão do modelo que deseja. Selecione a versão.
16. Selecione Editar endpoint para salvar.

Usando Trusted Advisor com o Amazon Comprehend

AWS Trusted Advisor é uma ferramenta on-line que fornece recomendações para ajudá-lo a provisionar seus recursos seguindo as AWS melhores práticas.

Se você tiver um plano Basic ou Developer Support, poderá usar o Trusted Advisor console para acessar todas as verificações na categoria Limites de Serviço e seis verificações na categoria Segurança. Se você tiver um plano de suporte Business ou Enterprise, poderá usar o Trusted Advisor console e a AWS Support API para acessar todas as Trusted Advisor verificações.

O Amazon Comprehend oferece suporte às Trusted Advisor seguintes verificações para ajudar os clientes a otimizar o custo e a segurança de seus endpoints do Amazon Comprehend, fornecendo recomendações práticas.

Endpoints do Amazon Comprehend subutilizados

A verificação dos endpoints subutilizados do Amazon Comprehend avalia a configuração do throughput dos seus endpoints. Essa verificação alerta quando os endpoints não são usados ativamente para solicitações de inferência em tempo real. Um endpoint que não é usado por mais de 15 dias é considerado subutilizado. Todos os endpoints acumulam cobranças com base tanto no conjunto de throughput, quanto no período em que o endpoint está ativo. Para endpoints não usados nos últimos 15 dias, recomendamos definir uma política de escalabilidade para o recurso usando Ajuste de escala automático para aplicativos. Para um endpoint que não foi usado nos últimos 30 dias e tem uma política de ajuste de escala automático definida, recomendamos que você use inferência assíncrona ou a exclua. Esses resultados de verificação são atualizados automaticamente uma vez por dia e podem ser visualizados na CostOptimizationcategoria no Trusted Advisor console.

Para visualizar o status de utilização de todos os seus endpoints e as recomendações correspondentes

1. Faça login no AWS Management Console e abra o Trusted Advisor console.
2. No painel de navegação, escolha a categoria de CostOptimizationverificação.
3. Na página de categoria, você pode visualizar o resumo de cada categoria de verificação:

- Ação recomendada (vermelho) — Trusted Advisor recomenda uma ação para a verificação.
- Investigação recomendada (amarelo): Trusted Advisor detecta um possível problema para a verificação.
- Nenhum problema detectado (verde) — Trusted Advisor não detecta um problema na verificação.
- Itens excluídos (cinza): O número de verificações que excluíram itens, como recursos que você deseja que uma verificação ignore.

4. Escolha a verificação de endpoints subutilizados do Amazon Comprehend para ver a descrição da verificação e os seguintes detalhes:

   - Critérios de alerta: descreve o limite quando uma verificação mudará o status.
   - Recommended Action (Ação recomendada) - Descreve as ações recomendadas para esta verificação.
   - Tabela de recursos: uma tabela que lista os detalhes do seu endpoint e o status de cada um com base em suas recomendações.

5. Na tabela de recursos, se um endpoint for sinalizado com uma investigação recomendada devido a um aviso de não usado nos últimos 30 dias, você poderá navegar até a página de detalhes do endpoint no console do Amazon Comprehend.

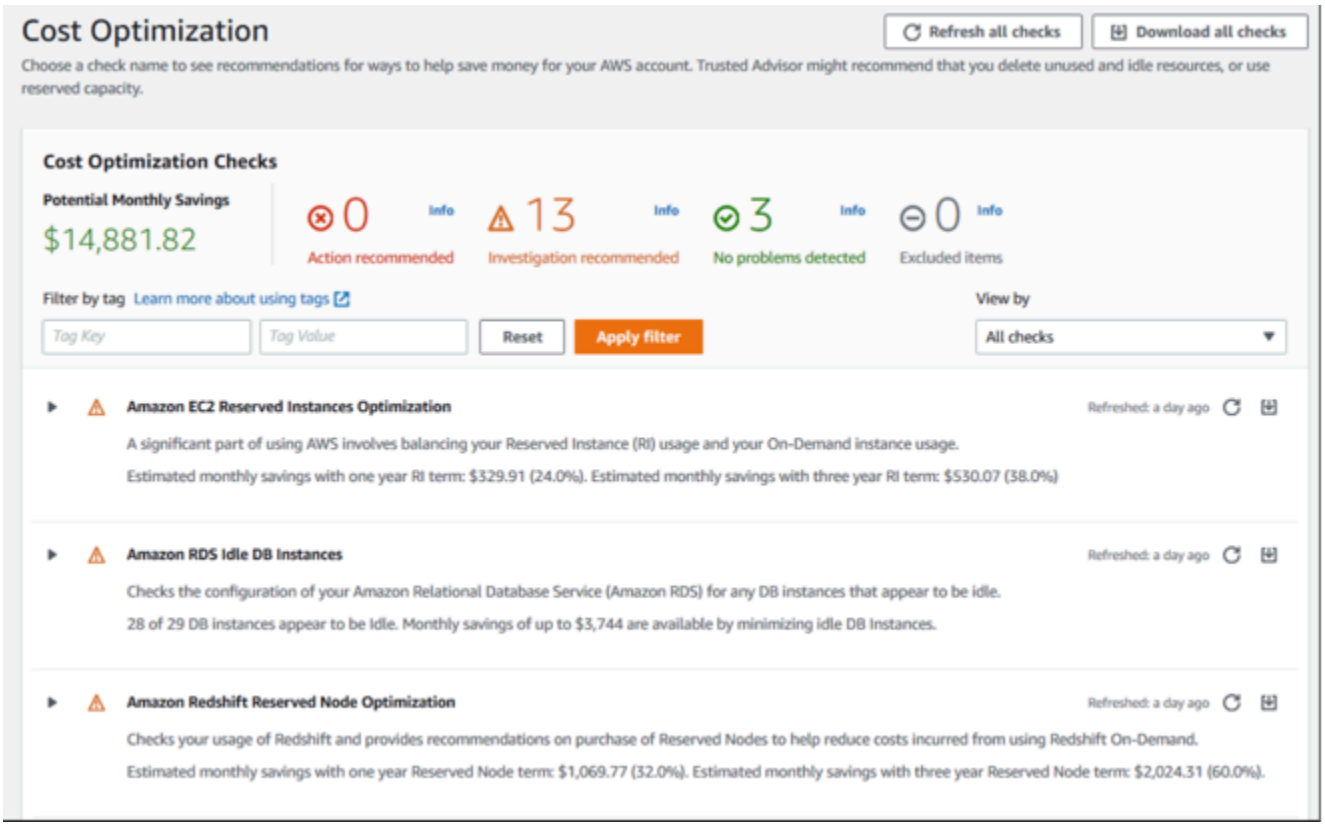
   - Se você não quiser mais usar esse endpoint, escolha Excluir.
   - Escolha Excluir para confirmar a exclusão. A página de detalhes do modelo personalizado é exibida. Confirme se o endpoint que você excluiu mostra excluindo ao lado dele. Depois de excluído, o endpoint é removido da lista de Endpoints.

6. Na tabela de recursos no Trusted Advisor console, se um endpoint estiver marcado com o status Investigation Recommended porque não foi usado nos últimos 15 dias e se tiver sido AutoScaling desativado, você poderá navegar até a página de detalhes do endpoint no console do Amazon Comprehend para ajustar o endpoint.

   - Se você quiser reduzir o throughput configurado para esse endpoint, clique em Editar Insira o número atualizado de unidades de inferência a serem atribuídas ao endpoint, marque a caixa de seleção para confirmar e escolha Editar Endpoint. Quando a atualização for concluída, o status será exibido como Pronto.
   - Se você quiser definir automaticamente o provisionamento de endpoints em seu endpoint em vez de ajustar manualmente a configuração da throughput, recomendamos usar o ajuste de escala automático dos aplicativos.

7. Na tabela de recursos no Trusted Advisor console, se um endpoint estiver marcado com o status Sem problemas detectados devido ao motivo Usado ativamente, isso significa que o endpoint está sendo utilizado ativamente para executar solicitações de inferência em tempo real e nenhuma ação é recomendada.

Aqui está um exemplo que mostra a visualização da CostOptimization categoria no Trusted Advisor console:



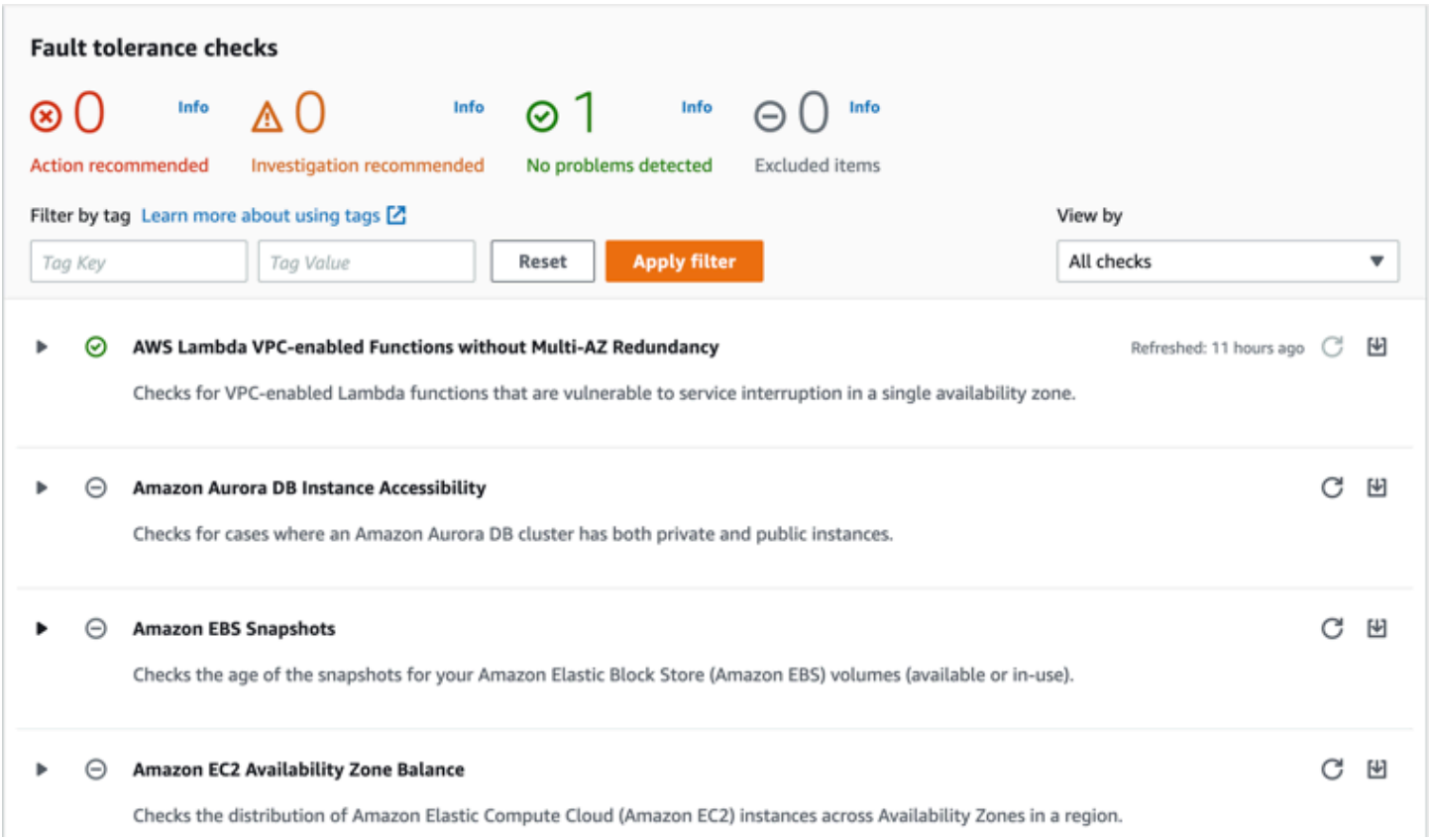
Risco de acesso ao endpoint do Amazon Comprehend

A verificação do risco de acesso ao endpoint do Amazon Comprehend avalia as permissões da chave do AWS Key Management Service (AWS KMS) para um endpoint em que o modelo subjacente foi criptografado usando chaves gerenciadas pelo cliente. Se a chave gerenciada pelo cliente estiver desabilitada, ou se a política de chave foi alterada para alterar as permissões permitidas para o Amazon Comprehend, a disponibilidade do endpoint pode ser afetada. Se a chave tiver sido desabilitada, recomendamos que você a habilite. Se a política de chaves foi alterada e você deseja continuar usando esse endpoint, recomendamos que você atualize a política de chaves. Os resultados da verificação são atualizados automaticamente várias vezes durante o dia. Essa verificação pode ser visualizada na categoria Tolerância a falhas do console do Trusted Advisor .

Para visualizar o status da AWS KMS chave de seus endpoints do Amazon Comprehend

1. Faça login no AWS Management Console e abra o Trusted Advisor console.
2. No painel de navegação, escolha a categoria de FaultToleranceverificação.
3. Na página de categoria, você pode visualizar o resumo de cada categoria de verificação:
   - Ação recomendada (vermelho) — Trusted Advisor recomenda uma ação para a verificação.
   - Investigação recomendada (amarelo): o Trusted Advisor detecta um possível problema para a verificação.
   - Nenhum problema detectado (verde) — Trusted Advisor não detecta um problema na verificação.
   - Itens excluídos (cinza): o número de verificações que excluíram itens, como recursos que você deseja que uma verificação ignore.
4. Escolha a verificação do risco de acesso ao endpoint do Amazon Comprehend e você poderá ver a descrição de verificação e os seguintes detalhes:
   - Critérios de alerta: descreve o limite quando uma verificação mudará o status.
   - Ação recomendada - descreve as ações recomendadas para esta verificação.
   - Tabela de recursos: uma tabela que lista os detalhes do endpoint criptografado do KMS e o status de cada um, com base na existência de ações recomendadas.
5. Na tabela Recursos, se um endpoint estiver marcado com o status Ação recomendada, selecione o link na KeyId coluna KMS e você será redirecionado para a página chave correspondente. AWS KMS
   - Para ativar uma AWS KMS chave desativada, escolha Ações principais e selecione Ativar.
   - Se o status da chave estiver listado como Habilitado, atualize a política de chave escolhendo Alternar para a visualização da política na seção Política de chave. Edite o documento da política de chaves para fornecer as permissões necessárias ao Amazon Comprehend e, em seguida, escolha Salvar alterações.

Aqui está um exemplo da visualização por FaultTolerance categoria no Trusted Advisor console:



Essas verificações e seus resultados também podem ser visualizados consultando a Trusted Advisor seção da AWS Support API.

Para saber mais sobre como configurar alarmes usando CloudWatch, consulte: Criando Trusted Advisor alarmes usando. CloudWatch Para obter um conjunto completo de verificações de Trusted Advisor melhores práticas, consulte: lista de verificação de AWS Trusted Advisor melhores práticas.

Excluindo endpoints do Amazon Comprehend

Quando você não precisar mais do seu endpoint, você deverá excluí-lo para parar de incorrer em custos com ele. Você pode criar facilmente outro endpoint sempre que precisar na seção Endpoints.

Para excluir um endpoint (console)

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. No menu à esquerda, escolha Endpoints.
3. Na Tabela de endpoints, localize o endpoint que você deseja excluir. Você pode pesquisar ou filtrar todos os endpoints para encontrar o que você precisa.

4. Marque a caixa de seleção para o endpoint que você deseja excluir. No canto superior direito da tabela de endpoints, selecione o ícone Ações.
5. Escolha Excluir.
6. Escolha Excluir para confirmar a exclusão. A página dos endpoints é exibida. Confirme se o endpoint que você excluiu mostra Excluindo ao lado dele. Quando ele é excluído, o endpoint é removido da lista de endpoints.

Para excluir um endpoint (AWS CLI)

O exemplo a seguir demonstra o uso da DeleteEndpointoperação com a AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws comprehend delete-endpoint \
    --endpoint-arn arn:aws:comprehend:region:account-id endpoint/endpoint name
```

Se a ação for executada com êxito, o Amazon Comprehend responderá com uma resposta HTTP 200 com um corpo HTTP vazio.

Ajuste de escala automático com endpoints

Em vez de ajustar manualmente o número de unidades de inferência provisionadas para seus endpoints de classificação de documentos e endpoints de reconhecimento de entidades, você pode usar o ajuste de escala automático para definir automaticamente o provisionamento de endpoints de acordo com suas necessidades de capacidade.

Há duas maneiras de usar o ajuste de escala automático para ajustar o número de unidades de inferência provisionadas para seu endpoint:

- Monitoramento do objetivo: defina o ajuste de escala automático para ajustar o provisionamento de endpoints de acordo com as necessidades de capacidade com base no uso.
- Escalabilidade programada: defina o ajuste de escala automático para ajustar o provisionamento de endpoints de acordo com as necessidades de capacidade em um cronograma especificado.

Você pode definir o escalonamento automático somente com o AWS Command Line Interface (AWS CLI). Para obter mais informações sobre o ajuste de escala automático, consulte O que é o Application Auto Scaling?

Monitoramento do objetivo

Com o rastreamento de destinos, você pode ajustar o provisionamento de endpoints para atender às suas necessidades de capacidade com base no uso. O número de unidades de inferência se ajusta automaticamente para que a capacidade utilizada esteja dentro de uma porcentagem alvo da capacidade provisionada. Você pode usar o rastreamento de destinos para acomodar picos temporários de uso de seus endpoints de classificação de documentos e endpoints de reconhecimento de entidades. Para obter mais informações, consulte Políticas de dimensionamento com monitoramento do objetivo para o Application Auto Scaling.

> **Note**
>
> Os exemplos a seguir estão formatados para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Configurar o rastreamento de destinos

Para configurar o rastreamento de destinos para um endpoint, você usa comandos da AWS CLI para registrar um destino escalável e, em seguida, criar uma política de escalabilidade. O destino escalável define as unidades de inferência como o recurso usado para ajustar o provisionamento de endpoints e a política de escalabilidade define as métricas que controlam o ajuste de escala automático da capacidade provisionada.

Para configurar o rastreamento de destinos

1. Registrar um destino escalável. Os exemplos a seguir registram um destino escalável para ajustar o provisionamento de endpoints com uma capacidade mínima de 1 unidade de inferência e uma capacidade máxima de 2 unidades de inferência.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

   ```
   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
```

Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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. Para verificar o registro do alvo escalável, use o seguinte comando da AWS CLI:

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace comprehend \
    --resource-id endpoint ARN
```

3. Crie uma configuração de rastreamento de destino para a política de escalabilidade e salve a configuração em um arquivo chamado `config.json`. Veja a seguir um exemplo de uma configuração de rastreamento de metas para um endpoint de classificação de documentos que visa manter a InferenceUtilization métrica em 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"
  }
}
```

Veja a seguir um exemplo de um endpoint reconhecedor de entidades:

```
{
  "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. Criar uma política de escalabilidade. Os exemplos a seguir criam uma política de escalabilidade com base na configuração de rastreamento de destino definida no arquivo `config.json`.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

   ```
   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
   ```

   Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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
```

Considerações

As seguintes considerações se aplicam ao usar o rastreamento de alvos com endpoints Comprehend:

- As métricas de endpoint são emitidas somente para solicitações bem-sucedidas. As métricas não aparecerão para solicitações que são limitadas ou falham devido a um erro interno do servidor ou do cliente.
- Quando faltam pontos de dados, os estados do CloudWatch alarme de apoio mudarão para`INSUFFICIENT_DATA` . Quando isso acontece, o Application Auto Scaling não pode escalar seu endpoint.
- A matemática métrica pode ser útil para contornar essa limitação. Por exemplo, para usar um valor de 0 quando nenhuma métrica for relatada, use a `FILL(m1,0)` função where `m1` is the metric. É importante testar sua configuração para garantir que ela esteja se comportando conforme o esperado. Consulte Criação de políticas de rastreamento de metas usando matemática métrica para obter mais opções.

Removendo o rastreamento de destinos

Para remover o rastreamento de destino de um endpoint, você usa comandos da AWS CLI para excluir a política de escalabilidade e, em seguida, cancelar o registro do destino escalável.

Para remover o rastreamento de destino

1. Exclua a política de escalabilidade Os exemplos a seguir excluem a política de escalabilidade especificada.

Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

```
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 \
```

Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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. Cancelar o registro de capacidade pretendida escalável. Os exemplos a seguir cancelam o registro de um destino escalável especificado.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

```
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
```

   Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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
```

Escalabilidade programada

Com a escalabilidade programada, você pode ajustar o provisionamento de endpoints para atender às suas necessidades de capacidade em um cronograma específico. A escalabilidade programada ajusta automaticamente o número de unidades de inferência para acomodar picos de uso em horários específicos. Você pode usar a escalabilidade programada para endpoints de classificação de documentos e endpoints de reconhecimento de entidades. Para mais informações sobre escalabilidade programada, consulte Escalabilidade programada para Ajuste de escala automático do aplicativo.

> **Note**
>
> Os exemplos a seguir estão formatados para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Configurar a escalabilidade programada

Para configurar o escalonamento agendado para um endpoint, você usa comandos da AWS CLI para registrar um destino escalável e, em seguida, criar uma ação agendada. O destino escalável define as unidades de inferência como o recurso usado para ajustar o provisionamento de endpoints e a ação programada controla o ajuste de escala automático da capacidade provisionada em horários específicos.

Para configurar a escalabilidade programada

1. Registrar um destino escalável. Os exemplos a seguir registram um destino escalável para ajustar o provisionamento de endpoints com uma capacidade mínima de 1 unidade de inferência e uma capacidade máxima de 2 unidades de inferência.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

   ```
   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
```

Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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. Criar uma ação programada. Os exemplos a seguir criam uma ação programada para ajustar automaticamente a capacidade provisionada todos os dias às 12:00 UTC com um mínimo de 2 unidades de inferência e um máximo de 5 unidades de inferência. Para obter mais informações sobre expressões cronológicas e escalabilidade programada, consulte Expressões de programação.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

```
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
```

   Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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
```

Removendo a escalabilidade programada

Para remover o escalonamento programado de um endpoint, você usa comandos AWS CLI para excluir a ação agendada e, em seguida, cancelar o registro do destino escalável.

Para remover a escalabilidade programada

1. Excluir as ações programadas. Os exemplos a seguir excluem uma ação programada específica.

   Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

   ```
   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
   ```

   Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

   ```
   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. Cancelar o registro de capacidade pretendida escalável. Os exemplos a seguir cancelam o registro de um destino escalável especificado.

Para um endpoint de classificação de documentos, use o seguinte comando da AWS CLI:

```
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
```

Para um endpoint reconhecedor de entidades, use o seguinte comando da CLI: AWS

```
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
```

Marcando seus Recursos

Uma tag é um par de chave/valor que pode ser adicionado a um recurso do Amazon Comprehend como metadados. É possível usar tags em tarefas de análise, modelos de classificação personalizada, modelos de reconhecimento de entidades personalizadas e endpoints. As tags têm duas funções principais: organizar seus recursos e fornecer controle de acesso baseado em tags.

Para organizar seus recursos com tags, adicione a chave de tag 'Departamento' e os valores de tag 'Vendas' ou 'Jurídico'. Em seguida, pesquise e filtre recursos que sejam pertinentes ao departamento jurídico da sua empresa.

Para fornecer controle de acesso baseado em tags, crie políticas do IAM com permissões baseadas em tags. Uma política pode permitir ou proibir uma operação com base nas tags fornecidas na sua solicitação (tags de solicitação) ou nas tags associadas ao recurso que está chamando (tags de recursos). Para mais informações sobre o uso de tags com o IAM, consulte Como controlar o acesso usando tags no Guia do usuário do IAM.

Considerações sobre o uso de tags com o Amazon Comprehend:

- Você pode adicionar até 50 tags por recurso, e elas podem ser adicionadas no momento da criação do recurso ou retroativamente.
- Uma chave de tag é um campo obrigatório, mas um valor de tag é opcional.
- As tags não precisam ser exclusivas entre os recursos, mas um determinado recurso não pode ter chaves de tag duplicadas.
- As chaves e valores das tags diferenciam maiúsculas de minúsculas.
- Uma chave de tag pode ter no máximo 127 caracteres; um valor de tag pode ter no máximo 255 caracteres.
- O prefixo `aws:` " está reservado para AWS uso; você não pode adicionar, editar ou excluir tags cuja chave comece com`aws:`. Essas tags não contam para seu tags-per-resource limite de 50.

> **Note**
>
> Se você planeja usar seu esquema de marcação em vários AWS serviços e recursos, lembre-se de que outros serviços podem ter requisitos diferentes para caracteres permitidos.

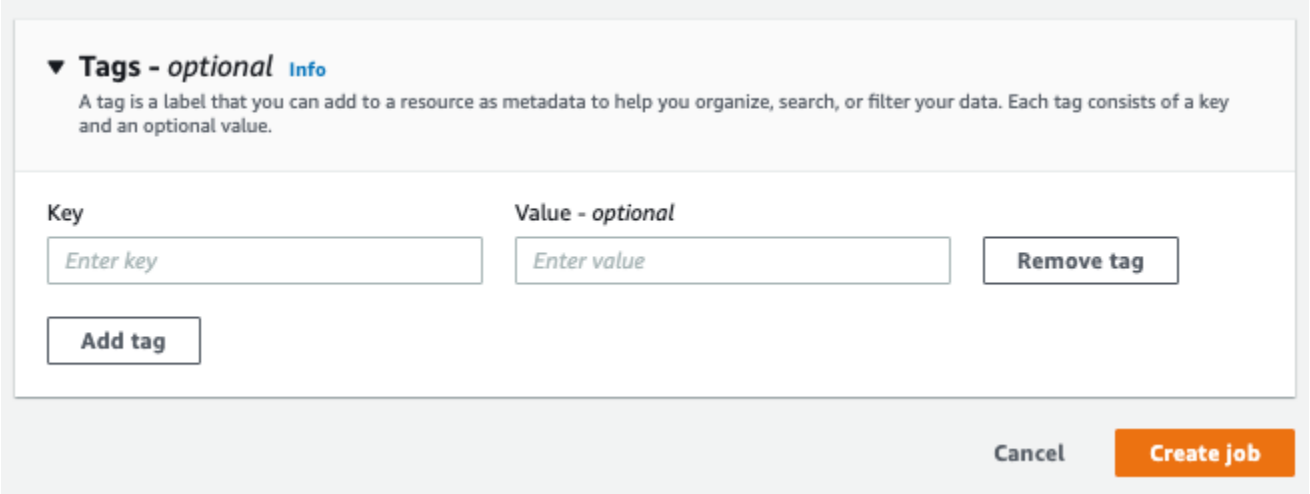
Tópicos

- Marcar um novo recurso
- Visualizar, editar e excluir tags associadas a um recurso

Marcar um novo recurso

Você pode adicionar tags a um Trabalho de análise, a um modelo de Classificação personalizada, a um modelo de Reconhecimento personalizado de entidade ou a endpoints.

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. Selecione o recurso (trabalho de análise, classificação personalizada ou reconhecimento personalizado de entidade) que deseja criar no painel de navegação esquerdo.
3. Clique em Criar trabalho (ou Criar novo modelo). Isso levará você para a página principal de "criação" para o recurso. Na parte inferior desta página, você verá o painel "Tags - opcionais".



   Insira uma chave de etiqueta e, opcionalmente, um valor de etiqueta. Escolha Adicionar tag para adicionar outra tag ao recurso. Repita esse processo até que todas as suas tags sejam adicionadas. Observe que as chaves de tags devem ser exclusivas por recurso.
4. Selecione o botão Criar ou Criar trabalho para continuar criando seu recurso.

Você também pode adicionar tags usando a AWS CLI. Este exemplo mostra como adicionar tags com o start-entities-detection-jobcomando.

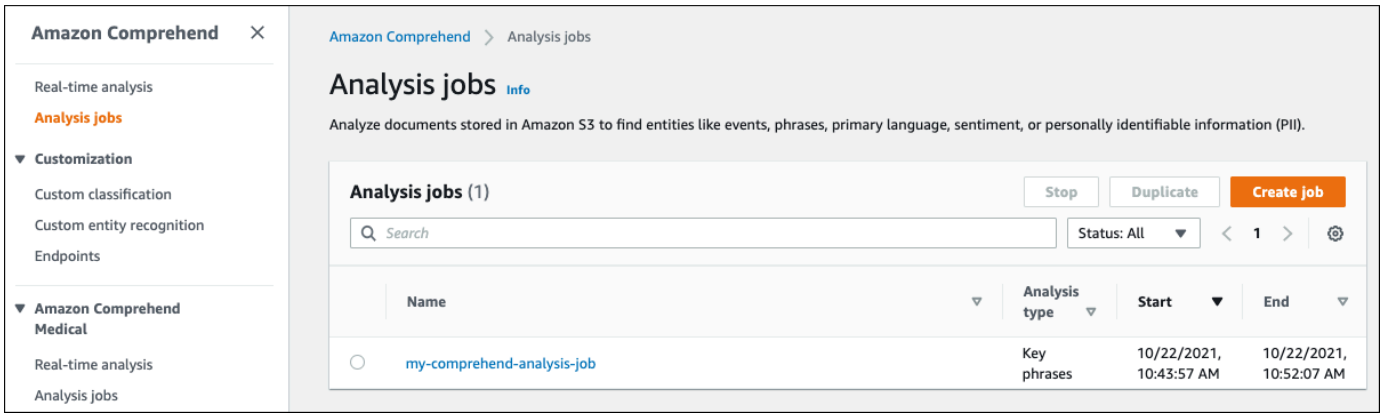
```
aws comprehend start-entities-detection-job \
```

```
--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\"}]"
```

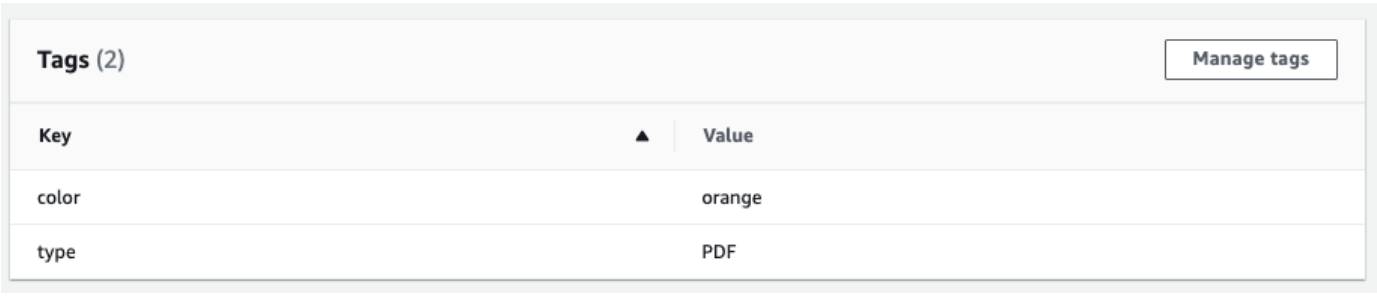
Visualizar, editar e excluir tags associadas a um recurso

Você pode visualizar as tags associadas a um Trabalho de análise, a um modelo de Classificação personalizada ou a um modelo de Reconhecimento personalizado de entidade.

1. Faça login no AWS Management Console e abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
2. Selecione o recurso (Trabalho de análise, Classificação personalizada ou Reconhecimento personalizado de entidade) que contém o arquivo com as tags que você deseja visualizar, modificar ou excluir. Isso exibirá a lista de arquivos existentes para o recurso selecionado.

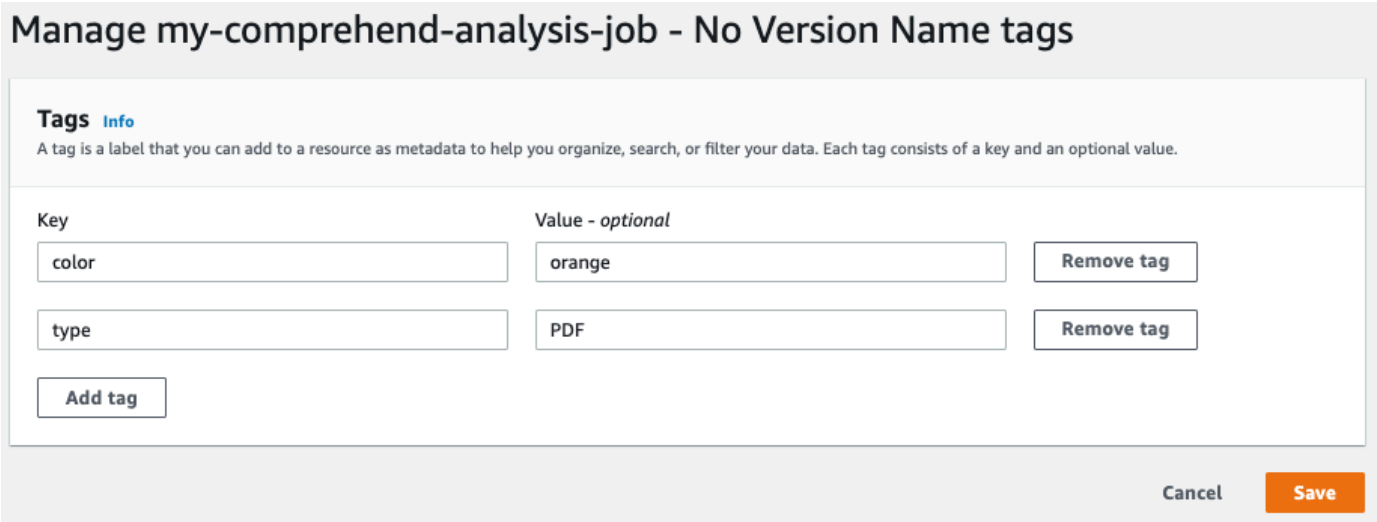


3. Clique no nome do arquivo (ou modelo) cujas tags deseja visualizar, modificar ou excluir. Você será redirecionado para a página de detalhes para aquele arquivo (ou modelo). Role para baixo até ver a caixa Tags. Nela, você poderá ver todas as tags associadas ao arquivo (ou modelo) selecionado.



Selecione Gerenciar tags para editar ou remover tags do seu recurso.

4. Clique no texto que deseja modificar e, em seguida, edite sua tag. Você também pode remover a tag selecionando Remover tag. Para adicionar uma nova tag, selecione Adicionar tag e insira o texto desejado nos campos em branco.



Quando terminar de modificar as tags, selecione Salvar.

Exemplos de código para o Amazon Comprehend usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon Comprehend com AWS um kit de desenvolvimento de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Exemplos básicos para o Amazon Comprehend usando AWS SDKs
  - Ações para o Amazon Comprehend usando AWS SDKs
    - Use CreateDocumentClassifier com um AWS SDK ou CLI
    - Use DeleteDocumentClassifier com um AWS SDK ou CLI
    - Use DescribeDocumentClassificationJob com um AWS SDK ou CLI
    - Use DescribeDocumentClassifier com um AWS SDK ou CLI
    - Use DescribeTopicsDetectionJob com um AWS SDK ou CLI
    - Use DetectDominantLanguage com um AWS SDK ou CLI
    - Use DetectEntities com um AWS SDK ou CLI
    - Use DetectKeyPhrases com um AWS SDK ou CLI
    - Use DetectPiiEntities com um AWS SDK ou CLI
    - Use DetectSentiment com um AWS SDK ou CLI
    - Use DetectSyntax com um AWS SDK ou CLI
    - Use ListDocumentClassificationJobs com um AWS SDK ou CLI
    - Use ListDocumentClassifiers com um AWS SDK ou CLI
    - Use ListTopicsDetectionJobs com um AWS SDK ou CLI

    - Use StartDocumentClassificationJob com um AWS SDK ou CLI
    - Use StartTopicsDetectionJob com um AWS SDK ou CLI
- Cenários para o Amazon Comprehend usando AWS SDKs
  - Criar uma aplicação de transmissão do Amazon Transcribe
  - Crie um chatbot Amazon Lex para engajar os visitantes do seu site
  - Crie um aplicativo web que envie e recupere mensagens usando o Amazon SQS
  - Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio
  - Detecte elementos do documento com o Amazon Comprehend e um SDK AWS
  - Detecte entidades no texto extraído de uma imagem usando um SDK AWS
  - Execute um trabalho de modelagem de tópicos do Amazon Comprehend em dados de amostra usando um SDK AWS
  - Treine um classificador Amazon Comprehend personalizado e classifique documentos usando um SDK AWS

Exemplos básicos para o Amazon Comprehend usando AWS SDKs

Os exemplos de código a seguir mostram como usar os conceitos básicos do Amazon Comprehend com. AWS SDKs

Exemplos

- Ações para o Amazon Comprehend usando AWS SDKs
  - Use CreateDocumentClassifier com um AWS SDK ou CLI
  - Use DeleteDocumentClassifier com um AWS SDK ou CLI
  - Use DescribeDocumentClassificationJob com um AWS SDK ou CLI
  - Use DescribeDocumentClassifier com um AWS SDK ou CLI
  - Use DescribeTopicsDetectionJob com um AWS SDK ou CLI
  - Use DetectDominantLanguage com um AWS SDK ou CLI
  - Use DetectEntities com um AWS SDK ou CLI
  - Use DetectKeyPhrases com um AWS SDK ou CLI
  - Use DetectPiiEntities com um AWS SDK ou CLI
  - Use DetectSentiment com um AWS SDK ou CLI

- Use DetectSyntax com um AWS SDK ou CLI
- Use ListDocumentClassificationJobs com um AWS SDK ou CLI
- Use ListDocumentClassifiers com um AWS SDK ou CLI
- Use ListTopicsDetectionJobs com um AWS SDK ou CLI
- Use StartDocumentClassificationJob com um AWS SDK ou CLI
- Use StartTopicsDetectionJob com um AWS SDK ou CLI

Ações para o Amazon Comprehend usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do Amazon Comprehend com. AWS SDKs Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Esses trechos chamam a API do Amazon Comprehend e são trechos de código de programas maiores que devem ser executados no contexto. É possível ver as ações em contexto em Cenários para o Amazon Comprehend usando AWS SDKs .

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência de API do Amazon Comprehend.

Exemplos

- Use CreateDocumentClassifier com um AWS SDK ou CLI
- Use DeleteDocumentClassifier com um AWS SDK ou CLI
- Use DescribeDocumentClassificationJob com um AWS SDK ou CLI
- Use DescribeDocumentClassifier com um AWS SDK ou CLI
- Use DescribeTopicsDetectionJob com um AWS SDK ou CLI
- Use DetectDominantLanguage com um AWS SDK ou CLI
- Use DetectEntities com um AWS SDK ou CLI
- Use DetectKeyPhrases com um AWS SDK ou CLI
- Use DetectPiiEntities com um AWS SDK ou CLI
- Use DetectSentiment com um AWS SDK ou CLI
- Use DetectSyntax com um AWS SDK ou CLI
- Use ListDocumentClassificationJobs com um AWS SDK ou CLI

- Use ListDocumentClassifiers com um AWS SDK ou CLI
- Use ListTopicsDetectionJobs com um AWS SDK ou CLI
- Use StartDocumentClassificationJob com um AWS SDK ou CLI
- Use StartTopicsDetectionJob com um AWS SDK ou CLI

Use **`CreateDocumentClassifier`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `CreateDocumentClassifier`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para criar um classificador de documentos para categorizar documentos

O exemplo de `create-document-classifier` a seguir inicia o processo de treinamento para um modelo de classificador de documentos. O arquivo de dados de treinamento, `training.csv`, está localizado na tag `--input-data-config`. `training.csv` é um documento de duas colunas em que os rótulos ou classificações são fornecidos na primeira coluna e os documentos são fornecidos na segunda coluna.

```
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
```

Saída:

```
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

Para obter mais informações, consulte Classificação personalizada no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte CreateDocumentClassifierem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
        }
    }
}
```

- Para obter detalhes da API, consulte CreateDocumentClassifiera Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a CreateDocumentClassifierReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteDocumentClassifier`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteDocumentClassifier`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para excluir um classificador de documentos personalizado

O exemplo de `delete-document-classifier` a seguir exclui um modelo de classificador de documentos personalizado.

```
aws comprehend delete-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Este comando não produz saída.

Para obter mais informações, consulte Gerenciar endpoints do Amazon Comprehend no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DeleteDocumentClassifierem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DeleteDocumentClassifierReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeDocumentClassificationJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeDocumentClassificationJob`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para descrever um trabalho de classificação de documentos

O exemplo de `describe-document-classification-job` a seguir obtém as propriedades de um trabalho assíncrono de classificação de documentos.

```
aws comprehend describe-document-classification-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
    "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"
    }
}
```

Para obter mais informações, consulte Classificação personalizada no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DescribeDocumentClassificationJobem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DescribeDocumentClassificationJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeDocumentClassifier`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeDocumentClassifier`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para descrever um classificador de documentos

O exemplo de `describe-document-classifier` a seguir obtém as propriedades de um modelo de classificador de documentos personalizado.

```
aws comprehend describe-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Saída:

```
{
    "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"
    }
}
```

Para obter mais informações, consulte Criação e gerenciamento de modelos personalizados no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DescribeDocumentClassifierem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DescribeDocumentClassifierReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeTopicsDetectionJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DescribeTopicsDetectionJob`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Execute uma tarefa de modelagem de tópicos em dados de amostra

CLI

AWS CLI

Para descrever um trabalho de detecção de tópicos

O exemplo de `describe-topics-detection-job` a seguir obtém as propriedades de um trabalho assíncrono de detecção de tópicos.

```
aws comprehend describe-topics-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
    "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"
    }
}
```

Para obter mais informações, consulte Análise assíncrona para insights do Amazon Comprehend no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DescribeTopicsDetectionJobem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DescribeTopicsDetectionJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DetectDominantLanguage`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectDominantLanguage`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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");
        }
    }
```

- Para obter detalhes da API, consulte DetectDominantLanguagea Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar o idioma dominante do texto de entrada

O `detect-dominant-language` a seguir analisa o texto de entrada e identifica o idioma dominante. A pontuação de confiança do modelo pré-treinado também é gerada.

```
aws comprehend detect-dominant-language \
```

```
    --text "It is a beautiful day in Seattle."
```

Saída:

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9877256155014038
        }
    ]
}
```

Para obter mais informações, consulte Idioma dominante no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DetectDominantLanguageem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
        }
    }
}
```

- Para obter detalhes da API, consulte DetectDominantLanguagea Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectDominantLanguageReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DetectEntities`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectEntities`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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");
        }
    }
```

- Para obter detalhes da API, consulte DetectEntitiesa Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar entidades nomeadas no texto de entrada

O exemplo de `detect-entities` a seguir analisa o texto de entrada e retorna as entidades nomeadas. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
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."
```

Saída:

```
{
    "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
        }
    ]
}
```

Para obter mais informações, consulte Entidades no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DetectEntitiesem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
```

```
        }
    }
}
```

- Para obter detalhes da API, consulte DetectEntitiesa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectEntitiesReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DetectKeyPhrases`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectKeyPhrases`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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");
        }
    }
```

- Para obter detalhes da API, consulte DetectKeyPhrasesa Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar frases-chave no texto de entrada

O exemplo de `detect-key-phrases` a seguir analisa o texto de entrada e identifica as principais frases nominais. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
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."
```

Saída:

```
{
    "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
        }
    ]
}
```

Para obter mais informações, consulte Frases-chave no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DetectKeyPhrasesem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
        }
    }
}
```

- Para obter detalhes da API, consulte DetectKeyPhrasesa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectKeyPhrasesReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DetectPiiEntities`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectPiiEntities`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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}");
                }
            }
        }
    }
```

- Para obter detalhes da API, consulte DetectPiiEntitiesa Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar entidades PII no texto de entrada

O exemplo de `detect-pii-entities` a seguir analisa o texto de entrada e identifica entidades que contêm informações de identificação pessoal (PII). A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
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."
```

Saída:

```
{
    "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
        }
    ]
}
```

Para obter mais informações, consulte Informações de identificação pessoal (PII) no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DetectPiiEntitiesem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectPiiEntitiesReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `DetectSentiment` com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectSentiment`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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");
        }
    }
```

- Para obter detalhes da API, consulte DetectSentimenta Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar o sentimento de um texto de entrada

O exemplo de `detect-sentiment` a seguir analisa o texto de entrada e retorna uma inferência do sentimento predominante (`POSITIVE`, `NEUTRAL`, `MIXED` ou `NEGATIVE`).

```
aws comprehend detect-sentiment \
    --language-code en \
    --text "It is a beautiful day in Seattle"
```

Saída:

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.9976957440376282,
        "Negative": 9.653854067437351e-05,
        "Neutral": 0.002169104292988777,
        "Mixed": 3.857641786453314e-05
    }
}
```

Para obter mais informações, consulte Sentimento no Guia do desenvolvedor do Amazon Comprehend

- Para obter detalhes da API, consulte DetectSentimentem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
        }
    }
}
```

- Para obter detalhes da API, consulte DetectSentimenta Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectSentimentReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DetectSyntax`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DetectSyntax`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Detecte elementos do documento

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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");
        }
    }
```

- Para obter detalhes da API, consulte DetectSyntaxa Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para detectar as partes da fala em um texto de entrada

O exemplo de `detect-syntax` a seguir analisa a sintaxe do texto de entrada e retorna as diferentes partes da fala. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

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

```
    --language-code en \
    --text "It is a beautiful day in Seattle."
```

Saída:

```
{
    "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
            }
        }
    ]
}
```

Para obter mais informações, consulte Análise de sintaxe no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte DetectSyntaxem Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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);
        }
    }
}
```

- Para obter detalhes da API, consulte DetectSyntaxa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a DetectSyntaxReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **ListDocumentClassificationJobs** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListDocumentClassificationJobs`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para listar trabalhos de classificação de documento

O exemplo de `list-document-classification-jobs` a seguir lista todos os trabalhos de classificação de documentos.

```
aws comprehend list-document-classification-jobs
```

Saída:

```
{
    "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"
        }
    ]
}
```

Para obter mais informações, consulte Classificação personalizada no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte ListDocumentClassificationJobsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a ListDocumentClassificationJobsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListDocumentClassifiers`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListDocumentClassifiers`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para listar todos os classificadores de documentos

O exemplo de `list-document-classifiers` a seguir lista todos os modelos de classificadores de documentos treinados e em treinamento.

```
aws comprehend list-document-classifiers
```

Saída:

```
{
    "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"
        }
    ]
}
```

Para obter mais informações, consulte Criação e gerenciamento de modelos personalizados no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte ListDocumentClassifiersem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a ListDocumentClassifiersReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListTopicsDetectionJobs`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListTopicsDetectionJobs`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Execute uma tarefa de modelagem de tópicos em dados de amostra

CLI

AWS CLI

Para listar todos os trabalhos de detecção de tópicos

O exemplo de `list-topics-detection-jobs` a seguir lista todos os trabalhos de detecção de tópicos assíncronos em andamento e concluídos.

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

Saída:

```
{
    "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"
        }
    ]
}
```

Para obter mais informações, consulte Análise assíncrona para insights do Amazon Comprehend no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte ListTopicsDetectionJobsem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a ListTopicsDetectionJobsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`StartDocumentClassificationJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `StartDocumentClassificationJob`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Treinar um classificador personalizado e classificar documentos

CLI

AWS CLI

Para iniciar um trabalho de classificação de documento

O exemplo de `start-document-classification-job` a seguir inicia um trabalho de classificação de documentos com um modelo personalizado em todos os arquivos no

endereço especificado pela tag `--input-data-config`. Neste exemplo, o bucket do S3 de entrada contém `SampleSMStext1.txt`, `SampleSMStext2.txt` e `SampleSMStext3.txt`. O modelo foi previamente treinado na classificação de documentos de mensagens de spam e não spam, ou "ham", e mensagens SMS. Quando o trabalho é concluído, `output.tar.gz` é colocado no local especificado pela tag `--output-data-config`. `output.tar.gz` contém `predictions.jsonl`, que lista a classificação de cada documento. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
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
```

Conteúdo de `SampleSMStext1.txt`:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Conteúdo de `SampleSMStext2.txt`:

```
"Hi, when do you want me to pick you up from practice?"
```

Conteúdo de `SampleSMStext3.txt`:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Saída:

```
{
    "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Conteúdo 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}]}
```

Para obter mais informações, consulte Classificação personalizada no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte StartDocumentClassificationJobem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a StartDocumentClassificationJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`StartTopicsDetectionJob`** com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `StartTopicsDetectionJob`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Execute uma tarefa de modelagem de tópicos em dados de amostra

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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}");
        }
    }
```

- Para obter detalhes da API, consulte StartTopicsDetectionJoba Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para iniciar um trabalho de análise de detecção de tópicos

O exemplo de `start-topics-detection-job` a seguir inicia um trabalho assíncrono de detecção de tópicos para todos os arquivos localizados no endereço especificado pela tag `--input-data-config`. Quando o trabalho é concluído, a pasta, `output`, é colocada no local especificado pela tag `--ouput-data-config`. `output` contém topic-terms.csv e doc-topics.csv. O primeiro arquivo de saída, topic-terms.csv, é uma lista de tópicos na coleção. Para cada tópico, a lista inclui por padrão os principais termos por tópico de acordo com seu peso. O segundo arquivo, `doc-topics.csv`, lista os documentos associados a um tópico e a proporção do documento relacionada ao tópico.

```
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
```

Saída:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

Para obter mais informações, consulte Modelagem de tópicos no Guia do desenvolvedor do Amazon Comprehend.

- Para obter detalhes da API, consulte StartTopicsDetectionJobem Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da 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
```

- Para obter detalhes da API, consulte a StartTopicsDetectionJobReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para o Amazon Comprehend usando AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Amazon AWS SDKs Comprehend com. Esses cenários mostram como realizar tarefas específicas chamando várias funções no Amazon Comprehend ou combinadas com outras. Serviços da AWS Cada cenário inclui um link para o código-fonte completo, onde podem ser encontradas instruções sobre como configurar e executar o código.

Os cenários têm como alvo um nível intermediário de experiência para ajudar você a compreender ações de serviço em contexto.

Exemplos

- Criar uma aplicação de transmissão do Amazon Transcribe
- Crie um chatbot Amazon Lex para engajar os visitantes do seu site
- Crie um aplicativo web que envie e recupere mensagens usando o Amazon SQS
- Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio
- Detecte elementos do documento com o Amazon Comprehend e um SDK AWS
- Detecte entidades no texto extraído de uma imagem usando um SDK AWS
- Execute um trabalho de modelagem de tópicos do Amazon Comprehend em dados de amostra usando um SDK AWS
- Treine um classificador Amazon Comprehend personalizado e classifique documentos usando um SDK AWS

Criar uma aplicação de transmissão do Amazon Transcribe

O exemplo de código a seguir mostra como construir uma aplicação que registra, transcreve e traduz áudio ao vivo em tempo real, e envia os resultados por e-mail.

JavaScript

SDK para JavaScript (v3)

Mostra como usar o Amazon Transcribe para construir uma aplicação que registra, transcreve e traduz áudio ao vivo em tempo real, e envia os resultados por e-mail usando o Amazon Simple Email Service (Amazon SES).

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Crie um chatbot Amazon Lex para engajar os visitantes do seu site

Os exemplos de código a seguir mostram como criar um chatbot para engajar os visitantes do seu site.

Java

SDK para Java 2.x

Mostra como usar a API do Amazon Lex para criar um Chatbot em uma aplicação da web para envolver os visitantes do seu site.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend

- Amazon Lex
- Amazon Translate

JavaScript

SDK para JavaScript (v3)

Mostra como usar a API do Amazon Lex para criar um Chatbot em uma aplicação da web para envolver os visitantes do seu site.

Para obter o código-fonte completo e instruções sobre como configurar e executar, consulte o exemplo completo Criando um chatbot Amazon Lex no guia do AWS SDK para JavaScript desenvolvedor.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Crie um aplicativo web que envie e recupere mensagens usando o Amazon SQS

Os exemplos de código a seguir mostram como criar uma aplicação de mensagens usando o Amazon SQS.

Java

SDK para Java 2.x

Mostra como usar a API do Amazon SQS para desenvolver uma API REST que envia e recupera mensagens.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Amazon SQS

Kotlin

SDK para Kotlin

Mostra como usar a API do Amazon SQS para desenvolver uma API REST que envia e recupera mensagens.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Amazon SQS

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio

Os exemplos de código a seguir mostram como criar uma aplicação que analisa os cartões de comentários dos clientes, os traduz do idioma original, determina seus sentimentos e gera um arquivo de áudio do texto traduzido.

.NET

SDK para .NET

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.

- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK para Java 2.x

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract

- Amazon Translate

JavaScript

SDK para JavaScript (v3)

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub. Os trechos a seguir mostram como o AWS SDK para JavaScript é usado nas funções do 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 };
};
```

Serviços utilizados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK para Ruby

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detecte elementos do documento com o Amazon Comprehend e um SDK AWS

O exemplo de código a seguir mostra como:

- Detecte idiomas, entidades e frases-chave em um documento.
- Detecte informações de identificação pessoal (PII) em um documento.
- Detecte o sentimento de um documento.

- Detectar elementos sintáticos em um documento.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Crie uma classe que envolva as ações do 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
```

Chame funções na classe wrapper para detectar entidades, frases e muito mais em um documento.

```
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)
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
  - DetectDominantLanguage
  - DetectEntities
  - DetectKeyPhrases
  - DetectPiiEntities
  - DetectSentiment
  - DetectSyntax

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Detecte entidades no texto extraído de uma imagem usando um SDK AWS

O exemplo de código a seguir mostra como usar o Amazon Comprehend para detectar entidades em texto extraído pelo Amazon Textract de uma imagem armazenada no Amazon S3.

Python

SDK para Python (Boto3)

Mostra como usar o AWS SDK para Python (Boto3) em um notebook Jupyter para detectar entidades no texto extraído de uma imagem. Este exemplo usa o Amazon Textract para extrair texto de uma imagem armazenada no Amazon Simple Storage Service (Amazon S3) e no Amazon Comprehend para detectar entidades no texto extraído.

Este exemplo é um caderno Jupyter e deve ser executado em um ambiente que possa hospedar blocos de anotações. Para obter instruções sobre como executar o exemplo usando o Amazon SageMaker AI, consulte as instruções em TextractAndComprehendNotebook.ipynb.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Execute um trabalho de modelagem de tópicos do Amazon Comprehend em dados de amostra usando um SDK AWS

O exemplo de código a seguir mostra como:

- Execute uma tarefa de modelagem de tópicos do Amazon Comprehend em dados de amostra.
- Informações sobre a tarefa.
- Extrair dados de saída do trabalho no Amazon S3.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Crie uma classe wrapper para chamar as ações de modelagem de tópicos do 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
```

Use a classe wrapper para executar uma tarefa de modelagem de tópicos e obter dados dela.

```
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)
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
  - DescribeTopicsDetectionJob
  - ListTopicsDetectionJobs
  - StartTopicsDetectionJob

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Treine um classificador Amazon Comprehend personalizado e classifique documentos usando um SDK AWS

O exemplo de código a seguir mostra como:

- Crie um classificador de vários rótulos do Amazon Comprehend.
- Treine o classificador em dados de amostra.
- Execute uma tarefa de classificação em um segundo conjunto de dados.
- Extrair dados de saída da tarefa do Amazon S3.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Crie uma classe wrapper para chamar as ações do classificador de documentos 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
```

Crie uma classe para ajudar na execução do cenário.

```
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
```

Treine um classificador em um conjunto de GitHub problemas com rótulos conhecidos e, em seguida, envie um segundo conjunto de GitHub problemas ao classificador para que eles possam ser rotulados.

```
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)
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência de API do AWS SDK para Python (Boto3).
  - CreateDocumentClassifier
  - DeleteDocumentClassifier
  - DescribeDocumentClassificationJob
  - DescribeDocumentClassifier
  - ListDocumentClassificationJobs
  - ListDocumentClassifiers
  - StartDocumentClassificationJob

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Comprehend com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon Comprehend

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada do descreve isso como a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Comprehend, AWS consulte Serviços em Escopo por Programa de Conformidade Serviços em Escopo por de Conformidade.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon Comprehend. Os tópicos a seguir mostram como configurar o Amazon Comprehend para atender aos seus objetivos de segurança e compatibilidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Amazon Comprehend.

Tópicos

- Proteção de dados no Amazon Comprehend
- Identity and Access Management (Gerenciamento de identidade e acesso) para o Amazon Comprehend
- Log de chamadas de API do Amazon Comprehend com o AWS CloudTrail
- Validação de conformidade para o Amazon Comprehend
- Resiliência no Amazon Comprehend
- Segurança da infraestrutura no Amazon Comprehend

Proteção de dados no Amazon Comprehend

O modelo de responsabilidade AWS compartilhada se aplica à proteção de dados no Amazon Comprehend. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon Comprehend ou Serviços da AWS outro usando o console, a API AWS CLI ou. AWS SDKs Quaisquer dados inseridos em tags ou em campos de texto

de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Tópicos

- Criptografia do KMS no Amazon Comprehend
- Prevenção contra o ataque do “substituto confuso” em todos os serviços
- Trabalhos de proteção usando uma nuvem privada virtual da Amazon
- Amazon Comprehend e endpoints da VPC de interface (AWS PrivateLink)

Criptografia do KMS no Amazon Comprehend

O Amazon Comprehend trabalha AWS Key Management Service com AWS KMS() para fornecer criptografia aprimorada para seus dados. O Amazon S3 já permite que você criptografe seus documentos de entrada ao criar uma análise de texto, modelagem de tópicos ou trabalho personalizado do Amazon Comprehend. A integração com AWS KMS permite criptografar os dados no volume de armazenamento das tarefas Start* e Create*, além de criptografar os resultados de saída das tarefas Start* usando sua própria chave KMS.

Para o AWS Management Console, o Amazon Comprehend criptografa modelos personalizados com sua própria chave KMS. Para o AWS CLI, o Amazon Comprehend pode criptografar modelos personalizados usando sua própria chave KMS ou uma chave gerenciada pelo cliente (CMK) fornecida.

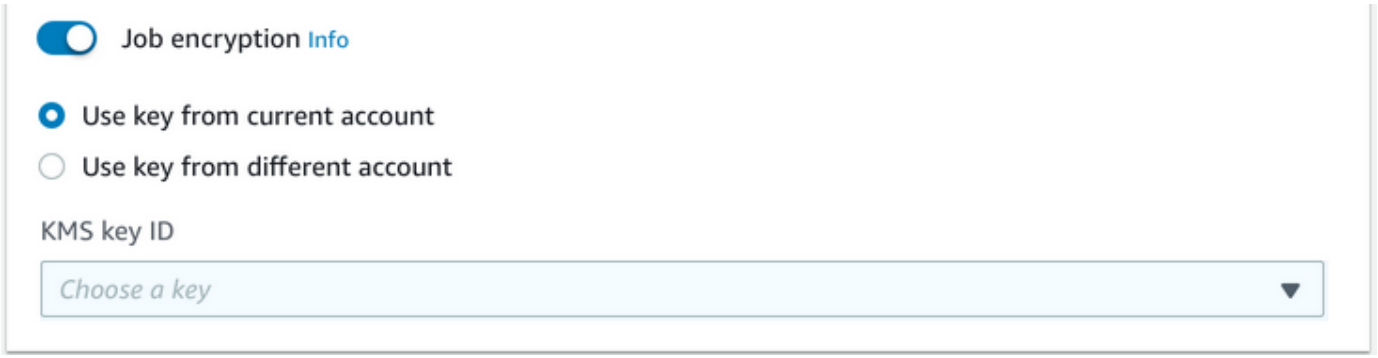
Criptografia KMS usando o AWS Management Console

Há duas opções de criptografia disponíveis ao usar o console:

- Criptografia de volumes
- Criptografia do resultado de saída

Para habilitar a criptografia de volume

1. Em Configurações do trabalho, escolha a opção Criptografia do trabalho.



2. Escolha se a chave gerenciada pelo cliente (CMK) KMS é da conta que você está usando no momento ou de uma conta diferente. Se desejar usar uma chave da conta atual, escolha a alias da chave de ID da chave KMS. Se estiver usando uma chave de uma conta diferente, você deverá inserir o ARN da chave.

Para habilitar a criptografia de resultado de saída

1. Em Configurações de saída, escolha a opção Criptografia.

2. Escolha se a chave gerenciada pelo cliente (CMK) é da conta que você está usando no momento ou de uma conta diferente. Se desejar usar uma chave da conta atual, escolha a chave de ID da chave KMS. Se estiver usando uma chave de uma conta diferente, você deverá inserir o ARN da chave.

Se você já tiver configurado a criptografia usando SSE-KMS nos documentos de entrada do S3, isso poderá fornecer segurança adicional. No entanto, se você fizer isso, o perfil do IAM usado deverá ter a permissão `kms:Decrypt` para a chave do KMS com a qual os documentos de entrada são criptografados. Para obter mais informações, consulte Permissões necessárias para usar a criptografia KMS.

Criptografia do KMS com operações de API

Todas as operações de API `Start*` e `Create*` do Amazon Comprehend são compatíveis com documentos de entrada criptografados do KMS. As operações de API `Describe*` e `List*` retornam o `KmsKeyId` no `OutputDataConfig` se o trabalho original tiver recebido `KmsKeyId` como entrada. Caso contrário, esse parâmetro não será retornado.

Isso pode ser visto no seguinte exemplo de AWS CLI usando a StartEntitiesDetectionJoboperação:

```
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**
>
> Este exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Criptografia de chave gerenciada pelo cliente (CMK) com operações de API

Operações de API de modelo personalizado do Amazon Comprehend,,, `CreateEndpoint` e `CreateEntityRecognizerCreateDocumentClassifier`, oferecem suporte à criptografia usando chaves gerenciadas pelo cliente por meio do. AWS CLI

É necessário usar uma política do IAM para permitir que uma entidade principal use ou gerencie chaves gerenciadas pelo cliente. Essas chaves são especificadas no elemento `Resource` da instrução de política. Como prática recomendada, limite as chaves gerenciadas pelo cliente somente àquelas que as entidades principais devem usar em sua declaração de política.

O exemplo de AWS CLI a seguir cria um reconhecedor de entidade personalizado com criptografia de modelo usando a operação: CreateEntityRecognizer

```
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**
>
> Este exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o Amazon Comprehend concede a outro serviço no recurso para o recurso. Se você utilizar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta `aws:SourceArn` no valor deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

A maneira mais eficaz de se proteger do problema ‘confused deputy’ é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou se especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:servicename::123456789012:*`.

Como usar a conta de origem

O exemplo a seguir mostra como usar a chave global de contexto de condição `aws:SourceAccount` no Amazon Comprehend.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
          "Service": "comprehend.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringEquals": {
             "aws:SourceAccount":"111122223333"
        }
    }
  }
}
```

Política de confiança para endpoints de modelos criptografados

Você precisa criar uma política de confiança para criar ou atualizar um endpoint para um modelo criptografado. Defina o valor `aws:SourceAccount` para o ID da conta. Se você usar a condição `ArnEquals`, defina o valor `aws:SourceArn` como o ARN do endpoint.

```
{
 "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"
            }
        }
```

```
    }
  ]
}
```

Criar modelo personalizado

Você precisa criar uma política de confiança para criar um modelo personalizado. Defina o valor `aws:SourceAccount` para o ID da sua conta. Se você usar a condição `ArnEquals`, defina o valor `aws:SourceArn` como o ARN da versão do modelo personalizado.

```
{
 "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"
              }
          }
      }
  ]
}
```

Trabalhos de proteção usando uma nuvem privada virtual da Amazon

O Amazon Comprehend usa uma variedade de medidas de segurança para garantir a segurança de seus dados com nossos contêineres de trabalho, onde eles são armazenados enquanto são usados pelo Amazon Comprehend. No entanto, os contêineres de trabalho acessam AWS recursos, como os buckets do Amazon S3, onde você armazena dados e artefatos de modelo, pela Internet.

Para controlar o acesso aos seus dados, recomendamos a criação de uma nuvem privada virtual (VPC) e a sua configuração para que os dados e os contêineres não sejam acessíveis pela internet. Para obter informações sobre como criar e configurar uma VPC, consulte Conceitos básicos da Amazon VPC no Guia do usuário da Amazon VPC. Usando uma VPC ajuda a proteger seus dados, pois é possível configurá-la para não se conectar à internet. Usar uma VPC também permite monitorar todo o tráfego de rede de entrada e saída de seus contêineres de trabalho, com os logs de fluxo da VPC. Para obter mais informações, consulte Logs de fluxo da VPC no Guia do usuário do Amazon Virtual Private Cloud.

Especifique a configuração de sua VPC ao criar um trabalho. Basta especificar as sub-redes e grupos de segurança. Quando você especifica as sub-redes e os grupos de segurança, o Amazon Comprehend cria interfaces de rede elásticas ENIs () associadas aos seus grupos de segurança em uma das sub-redes. ENIs permita que nossos contêineres de trabalho se conectem aos recursos em sua VPC. Para obter informações sobre isso ENIs, consulte Elastic Network Interfaces no Guia do usuário da Amazon VPC.

> **Note**
>
> Para trabalhos, você pode configurar apenas sub-redes com uma VPC de locação padrão em que sua instância é executada em hardware compartilhado. Para obter mais informações sobre o atributo de locação para VPCs, consulte Instâncias dedicadas no Guia do EC2 usuário da Amazon.

Configurar um trabalho para acesso ao Amazon VPC

Para especificar sub-redes e grupos de segurança na VPC, use o parâmetro da solicitação `VpcConfig` da API aplicável ou forneça essas informações ao criar um trabalho no console do Amazon Comprehend. O Amazon Comprehend usa essas informações para ENIs criá-las e anexá-las aos nossos contêineres de trabalho. Eles ENIs fornecem aos nossos contêineres de trabalho uma conexão de rede em sua VPC que não está conectada à Internet.

O seguinte APIs contém o parâmetro de `VpcConfig` solicitação:

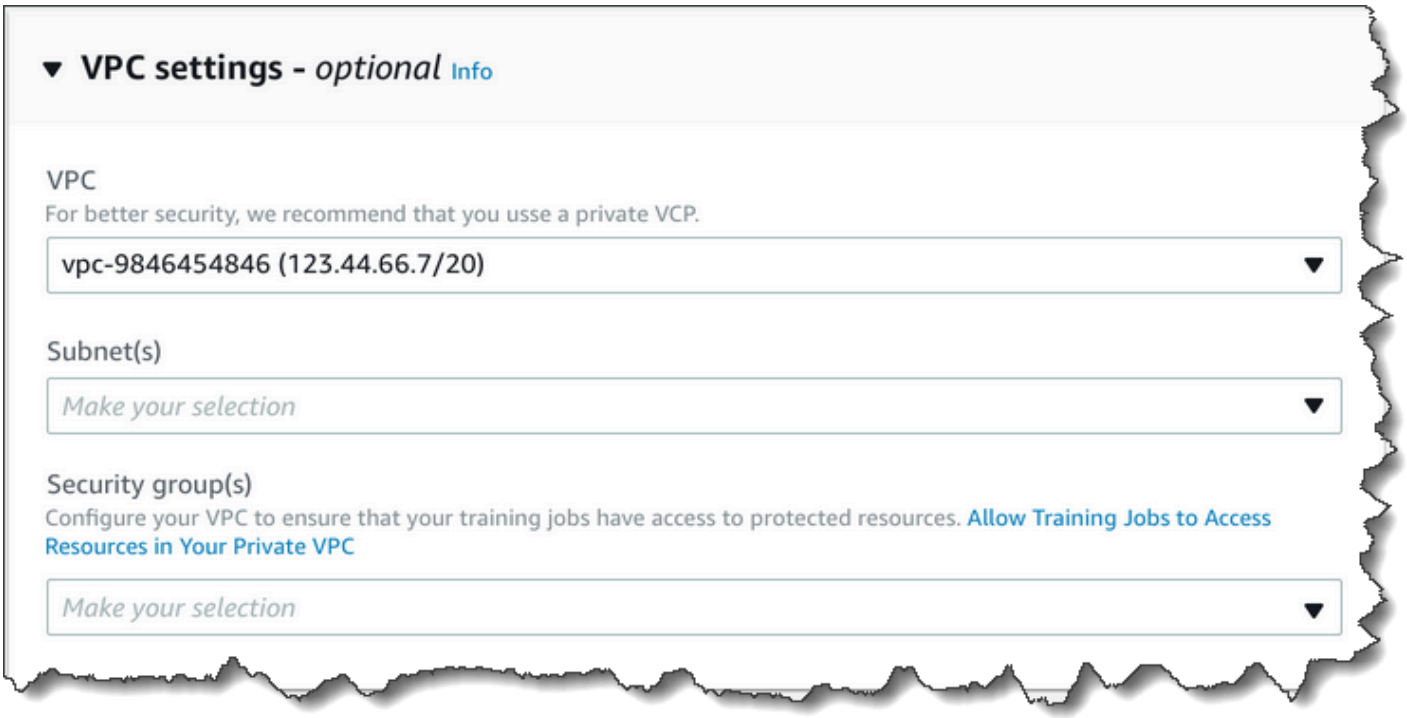
- `Create*` APIs: `CreateDocumentClassifier`, `CreateEntityRecognizer`
- `Start*` APIs: `StartDocumentClassificationJob`, `StartDominantLanguageDetectionJob`, `StartEntitiesDetectionJob`,

StartKeyPhrasesDetectionJob, StartSentimentDetectionJob, StartTargetedSentimentDetectionJob, StartTopicsDetectionJob

Veja a seguir um exemplo do VpcConfig parâmetro que você inclui na sua chamada de API:

```
"VpcConfig": {
      "SecurityGroupIds": [
          " sg-0123456789abcdef0"
          ],
      "Subnets": [
          "subnet-0123456789abcdef0",
          "subnet-0123456789abcdef1",
          "subnet-0123456789abcdef2"
          ]
      }
```

Para configurar uma VPC a partir do console do Amazon Comprehend, escolha os detalhes de configuração na seção opcional Configurações da VPC ao criar o trabalho.



Configurar a VPC para trabalhos do Amazon Comprehend

Ao configurar a VPC para os seus trabalhos de treinamento do Amazon Comprehend, use as diretrizes a seguir. Para obter informações sobre como configurar uma VPC, consulte Trabalho com VPCs e sub-redes no Guia do usuário da Amazon VPC.

Garanta que as sub-redes tenham endereços IP suficientes

Suas sub-redes da VPC devem ter pelo menos dois endereços IP privados para cada instância em um trabalho. Para obter mais informações, consulte Dimensionamento de VPC e sub-rede no Guia do usuário IPv4 da Amazon VPC.

Crie um endpoint da VPC do Amazon S3

Se você configurar sua VPC para que os contêineres de trabalho não tenham acesso à internet, eles não poderão se conectar aos buckets do Amazon S3 que contêm os dados, a menos que você crie um endpoint da VPC que permita o acesso. Ao criar um VPC endpoint, você permite que os contêineres de trabalho acessem seus dados durante os trabalhos de treinamento e análise.

Ao criar o VPC endpoint, configure estes valores:

- Selecione a categoria de serviço como AWS Serviços
- Especifique o serviço como `com.amazonaws.`*`region`*`.s3`
- Selecione Gateway como o tipo de VPC Endpoint

Se você estiver usando AWS CloudFormation para criar o VPC endpoint, siga a documentação. AWS CloudFormation VPCEndpoint O exemplo a seguir mostra a VPCEndpointconfiguração em um AWS CloudFormation modelo.

```
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
```

Recomendamos que você também crie uma política personalizada para que apenas solicitações da sua VPC acessem os buckets do S3. Para obter mais informações, consulte Endpoints para o Amazon S3 no Manual do usuário da Amazon VPC.

A política a seguir permite acesso aos buckets do S3. Edite essa política para que esse acesso seja permitido somente para os recursos de que seu trabalho precisa.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:DeleteObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

Use as configurações de DNS padrão para sua tabela de rotas de endpoints, para que o Amazon URLs S3 padrão (por exemplo`http://s3-aws-region.amazonaws.com/amzn-s3-demo-bucket`,) resolva. Se você não usar as configurações de DNS padrão, certifique-se de que as URLs que você usa para especificar os locais dos dados em seus trabalhos sejam resolvidas configurando as tabelas de rotas do endpoint. Para obter informações sobre as tabelas de rotas de endpoints da VPC, consulte Roteamento para endpoints de gateway no Guia do usuário da Amazon VPC.

A política de endpoint padrão permite que os usuários instalem pacotes dos repositórios do Amazon Linux e do Amazon Linux 2 em nosso contêiner de trabalhos. Se você não deseja que os usuários instalem pacotes, crie uma política de endpoint personalizada que negue explicitamente o acesso a esses repositórios. O Comprehend em si não precisa de nenhum desses pacotes, portanto, não haverá nenhum impacto na funcionalidade. Veja a seguir um exemplo de política que nega acesso somente a esses repositórios:

```
{
    "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/*"
                ]
            }
    ]
}
```

Permissões para a **`DataAccessRole`**

Quando você usa uma VPC com trabalho de análise, a `DataAccessRole` usado para as operações `Create*` e `Start*` também deve ter permissões para a VPC a partir da qual os documentos de entrada e o bucket de saída são acessados.

A política a seguir fornece o acesso necessário à `DataAccessRole` usada para nas operações `Create*` e `Start*`.

```
{
    "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": "*"
        }
    ]
}
```

Configuração do grupo de segurança da VPC

Com os trabalhos distribuídos, é necessário permitir a comunicação entre os diferentes contêineres de trabalho no mesmo trabalho. Para fazer isso, configure uma regra para seu grupo de segurança

que permita conexões de entrada entre membros do mesmo grupo de segurança. Para obter informações, consulte Regras de grupos de segurança no Guia do usuário da Amazon VPC.

Conecte-se a recursos fora de sua VPC

Se você configurar a VPC para que não tenha acesso à internet, os trabalhos que a utilizam não terão acesso a recursos externos. Se os seus trabalhos precisarem acessar recursos fora da VPC, conceda o acesso com uma das seguintes opções:

- Se seu trabalho precisar acessar um AWS serviço que ofereça suporte a endpoints VPC de interface, crie um endpoint para se conectar a esse serviço. Para obter uma lista dos serviços compatíveis com endpoints de interface, consulte Endpoint de VPCs no Guia do usuário da Amazon VPC. Para obter informações sobre a criação de uma interface VPC endpoint, consulte Interface VPC Endpoints () no Guia do usuário AWS PrivateLink da Amazon VPC.
- Se seu trabalho precisar acessar um AWS serviço que não ofereça suporte a endpoints VPC de interface ou a um recurso externo AWS, crie um gateway NAT e configure seus grupos de segurança para permitir conexões de saída. Para obter informações sobre como configurar um gateway NAT para sua VPC, consulte Cenário 2: VPC com sub-redes pública e privada (NAT) no Guia do usuário da Amazon VPC.

Amazon Comprehend e endpoints da VPC de interface (AWS PrivateLink)

É possível estabelecer uma conexão privada entre a VPC e o Amazon Comprehend criando um endpoint da VPC de interface. Os endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que permite acessar o Amazon APIs Comprehend de forma privada sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão Direct Connect. AWS As instâncias na sua VPC não precisam de endereços IP públicos para se comunicar com o Amazon Comprehend. APIs O tráfego de rede entre a VPC e o Amazon Comprehend não sai da rede da Amazon.

Cada endpoint de interface é representado por uma ou mais interfaces de rede elástica em suas sub-redes.

Para obter mais informações, consulte Endpoints da VPC da interface (AWS PrivateLink) no Manual do Usuário do Amazon VPC.

Considerações sobre endpoints da VPC do Amazon Comprehend

Antes de configurar um endpoint da VPC de interface para o Amazon Comprehend, revise as Propriedades e limitações do endpoint de interface no Guia do usuário do Amazon VPC.

Os endpoints do Amazon Comprehend não estão disponíveis em todas as zonas de disponibilidade de uma região. Ao criar o endpoint, use o comando a seguir para listar as zonas de disponibilidade.

```
aws ec2 describe-vpc-endpoint-services \
  --service-names com.amazonaws.us-west-2.comprehend
```

O Amazon Comprehend é compatível com chamadas para todas as ações de API da VPC.

Criar um endpoint da VPC de interface para o Amazon Comprehend

Você pode criar um VPC endpoint para o serviço Amazon Comprehend usando o console Amazon VPC ou o (). AWS Command Line Interface AWS CLI Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Crie um endpoint da VPC para o Amazon Comprehend usando o seguinte nome de serviço:

- com.amazonaws. *region*.compreender

Se você habilitar o DNS privado para o endpoint, poderá fazer solicitações de API para o Amazon Comprehend usando seu nome DNS padrão para a região, por exemplo, *comprehend.us-east-1.amazonaws.com*.

Para mais informações, consulte Acessar um serviço por um endpoint de interface no Guia do usuário da Amazon VPC.

Criar uma política de endpoint da VPC para o Amazon Comprehend

É possível anexar uma política de endpoint ao endpoint da VPC que controla o acesso ao Amazon Comprehend. Essa política especifica as seguintes informações:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos aos quais as ações podem ser aplicadas.

Para obter mais informações, consulte Controlar o acesso a serviços com endpoints da VPC no Guia do Usuário do Amazon VPC.

Exemplo: política de endpoint da VPC para ações do Amazon Comprehend

Veja a seguir um exemplo de política de endpoint para o Amazon Comprehend. Quando anexada a um endpoint, essa política concede acesso à ação `DetectEntities` do Amazon Comprehend para todas as entidades principais em todos os recursos.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "comprehend:DetectEntities"
         ],
         "Resource":"*"
      }
   ]
}
```

Identity and Access Management (Gerenciamento de identidade e acesso) para o Amazon Comprehend

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para utilizar os recursos do Amazon Comprehend. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Público
- Autenticar com identidades
- Gerenciar o acesso usando políticas
- Como o Amazon Comprehend funciona com o IAM
- Exemplos de políticas baseadas em identidade para o Amazon Comprehend
- AWS políticas gerenciadas para o Amazon Comprehend
- Solução de problemas de identidade e acesso do Amazon Comprehend

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no Amazon Comprehend.

Usuário do serviço: se você usar o serviço do Amazon Comprehend para fazer sua tarefa, o administrador fornecerá as credenciais e as permissões necessárias. À medida você que usar mais atributos do Amazon Comprehend para fazer sua tarefa, poderá precisar de permissões adicionais. Compreenda como o acesso é gerenciado pode ajudar a solicitar as permissões corretas ao administrador. Se não for possível acessar um atributo no Amazon Comprehend, consulte Solução de problemas de identidade e acesso do Amazon Comprehend.

Administrador do serviço: se você for o responsável pelos recursos do Amazon Comprehend na empresa, provavelmente terá acesso total a esse serviço. Cabe a você determinar quais funcionalidades e recursos do Amazon Comprehend os usuários do serviço deverão acessar. Envie as solicitações ao administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com o Amazon Comprehend, consulte Como o Amazon Comprehend funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez deseje saber detalhes sobre como gravar políticas para gerenciar o acesso ao Amazon Comprehend. Para visualizar exemplos de políticas baseadas em identidade do Amazon Comprehend que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade para o Amazon Comprehend.

Autenticar com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte Versão 4 do AWS Signature para solicitações de API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator da AWS no IAM no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte O que é o Centro de Identidade do IAM? no Guia do Usuário do AWS IAM Identity Center .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode alternar de um usuário para uma função do IAM (console). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Métodos para assumir um perfil no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada,

essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center .

- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.
  - Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar

uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível ao armazenamento de chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas

substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amazon Comprehend funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon Comprehend, saiba quais atributos do IAM estão disponíveis para uso com o Amazon Comprehend.

Atributos do IAM possíveis de usar com o Amazon Comprehend

Atributo do IAM	Suporte do Amazon Comprehend
Políticas baseadas em identidade	Sim
Políticas baseadas em atributos	Sim
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Sim
Perfis de serviço	Sim

Atributo do IAM	Suporte do Amazon Comprehend
Perfis vinculados a serviço	Não

Para ter uma visão de alto nível de como o Amazon Comprehend AWS e outros serviços funcionam com a maioria dos recursos do IAM AWS , consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade do Amazon Comprehend

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para o Amazon Comprehend

Para ver exemplos de políticas baseadas em identidade do Amazon Comprehend, consulte Exemplos de políticas baseadas em identidade para o Amazon Comprehend.

Políticas baseadas em recursos no Amazon Comprehend

Compatível com políticas baseadas em recursos: sim

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal

especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

O serviço Amazon Comprehend é compatível com apenas um tipo de política baseada em recurso (uma política de modelo personalizado), anexada a um modelo personalizado. Essa política define outras contas que podem usar o modelo personalizado.

Para saber como anexar uma política baseada em recursos a um modelo personalizado, consulte Políticas baseadas em recursos para modelos personalizados.

Ações de políticas para o Amazon Comprehend

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do Amazon Comprehend, consulte Ações definidas pelo Amazon Comprehend na Referência de autorização do serviço.

As ações de políticas no Amazon Comprehend usam o seguinte prefixo antes da ação:

```
comprehend
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "comprehend:DetectSentiment",
      "comprehend:ClassifyDocument"
         ]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "comprehend:Describe*"
```

Não use curingas para especificar todas as ações de um serviço. Use as melhores práticas de concessão de privilégio mínimo ao especificar as permissões usadas em uma política.

Para ver exemplos de políticas baseadas em identidade do Amazon Comprehend, consulte Exemplos de políticas baseadas em identidade para o Amazon Comprehend.

Recursos de políticas para o Amazon Comprehend

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do Amazon Comprehend e ARNs seus, consulte Recursos definidos pelo Amazon Comprehend na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon Comprehend.

Chaves de condição de políticas do Amazon Comprehend

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do Amazon Comprehend, consulte Chaves de condição do Amazon Comprehend na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar a chave de condição, consulte Ações definidas pelo Amazon Comprehend.

Para ver exemplos de políticas baseadas em identidade do Amazon Comprehend, consulte Exemplos de políticas baseadas em identidade para o Amazon Comprehend.

ACLs no Amazon Comprehend

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Amazon Comprehend

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define as permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Para ter mais informações sobre recursos de marcação do Amazon Comprehend, consulte Marcando seus Recursos .

Utilizar credenciais temporárias com o Amazon Comprehend

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS "Trabalhe com o IAM" no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil do IAM (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM.

Encaminhar sessões de acesso do Amazon Comprehend

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço do Amazon Comprehend

Compatível com perfis de serviço: sim

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon Comprehend. Edite perfis de serviço somente quando o Amazon Comprehend der orientação para isso.

Para usar as operações assíncronas do Amazon Comprehend, você deve conceder a ele o acesso ao bucket do Amazon S3 contendo sua coleção de documentos. Você faz isso criando um perfil de acesso a dados em sua conta com uma política de confiança para confiar na entidade principal do serviço do Amazon Comprehend.

Para ver um exemplo de política, consulte Permissões baseadas em perfis necessárias para operações assíncronas

Funções vinculadas ao serviço para o Amazon Comprehend

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para o Amazon Comprehend

Por padrão, usuários e perfis não têm permissão para criar nem modificar recursos do Amazon Comprehend. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Amazon Comprehend, incluindo o formato de cada um ARNs dos tipos de recursos, consulte Ações, recursos e chaves de condição para o Amazon Comprehend na Referência de Autorização de Serviço.

Tópicos

- Práticas recomendadas de política
- Utilizar o console do Amazon Comprehend
- Permitir que os usuários visualizem suas próprias permissões
- Permissões necessárias para executar ações de análise de documentos
- Permissões necessárias para usar a criptografia KMS
- AWS políticas gerenciadas (predefinidas) para o Amazon Comprehend
- Permissões baseadas em perfis necessárias para operações assíncronas
- Conceder permissões a todas as ações do Amazon Comprehend
- Permissões para ações de modelagem de tópicos
- Permissões necessárias para uma tarefa de análise assíncrona personalizada

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon Comprehend em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e passe para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também

conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.

- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Utilizar o console do Amazon Comprehend

Para acessar o console da Amazon Comprehend, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon Comprehend em seu. Conta da AWS Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para obter permissões mínimas do console Amazon Comprehend, você pode anexar *ComprehendReadOnly* AWS a política gerenciada às entidades. Para obter informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Para usar o console do Amazon Comprehend, também é necessário conceder permissões para as ações exibidas na política a seguir:

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
      "Action": [
          "iam:ListRoles",
          "iam:GetRole",
          "s3:ListAllMyBuckets",
          "s3:ListBucket",
          "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "*"
  }
  ]
  }
```

O console do Amazon Comprehend precisa dessas permissões adicionais pelas seguintes razões:

- Permissões do `iam` para listar os perfis do IAM disponíveis para sua conta.
- Permissões do `s3` para acessar os buckets e objetos do Amazon S3 contendo os dados para modelagem de tópicos.

Ao criar uma tarefa em lote assíncrono ou uma tarefa de modelagem de tópicos usando o console, você tem a opção de fazer com que o console crie um perfil do IAM para sua tarefa. Para criar um perfil do IAM, os usuários devem receber as seguintes permissões adicionais para criar políticas e perfis do IAM e anexar políticas aos perfis:

```
{
  "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*"
      }
    ]
  }
```

O console do Amazon Comprehend precisa dessas permissões adicionais pelas seguintes razões:

- Permissões do `iam` para criar e anexar perfis e políticas. A ação `iam:PassRole` permite que o console passe o perfil para o Amazon Comprehend.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
}
```

Permissões necessárias para executar ações de análise de documentos

O exemplo de política a seguir concede permissões para usar as ações de análise de documentos do 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": "*"
    }
 ]
}
```

A política tem uma instrução que concede permissão para usar as ações `DetectEntities`, `DetectKeyPhrases`, `DetectDominantLanguage`, `DetectTargetedSentiment`, `DetectSentiment`, e `DetectSyntax`. A declaração de política também concede permissões para usar dois métodos da API Amazon Textract. O Amazon Comprehend chama esses métodos para extrair texto de arquivos de imagem e documentos PDF digitalizados. É possível remover essas permissões para usuários que nunca executam inferência personalizada para esses tipos de arquivos de entrada.

Um usuário com essa política não conseguiria realizar ações em lote ou ações assíncronas em sua conta.

A política não especifica o elemento `Principal`, porque não se especifica o principal que obtém as permissões em uma política baseada em identidade. Quando você anexar uma política a um usuário, o usuário será a entidade principal implícita. Quando você anexa uma política de permissões a um perfil do IAM, o principal identificado na política de confiança do perfil obtém as permissões.

Para obter uma tabela que mostra todas as ações da API do Amazon Comprehend e os recursos aos quais elas se aplicam, consulte Ações, recursos e chaves de condição do Amazon Comprehend na Referência de autorização de serviço.

Permissões necessárias para usar a criptografia KMS

Para usar totalmente o Amazon Key Management Service (KMS) para criptografia de dados e tarefas em uma tarefa assíncrona, é preciso conceder permissões para as ações mostradas na seguinte política:

```
{
  "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"
                ]
            }
        }
    }
  ]
}
```

Ao criar uma tarefa assíncrona com o Amazon Comprehend, você usa dados de entrada armazenados no Amazon S3. Com o S3, você tem a opção de criptografar seus dados armazenados, os quais são criptografados pelo S3, não pelo Amazon Comprehend. Podemos descriptografar e ler esses dados de entrada criptografados se você fornecer ao `kms:Decrypt` permissão para a chave com a qual os dados de entrada originais foram criptografados para o perfil de acesso a dados usada pela tarefa do Amazon Comprehend.

Você também tem a opção de usar chaves gerenciadas pelo cliente (CMK) do KMS para criptografar os resultados de saída no S3, bem como o volume de armazenamento usado durante o processamento da tarefa. Ao fazer isso, é possível usar a mesma chave KMS para os dois tipos de criptografia, mas não é necessário. Campos separados estão disponíveis ao criar a tarefa a fim de especificar as chaves para criptografia de saída e criptografia de volume, e você pode até mesmo usar uma chave KMS de uma conta diferente.

Ao usar a criptografia KMS, é necessária a permissão `kms:CreateGrant` para criptografia de volume e a permissão de `kms:GenerateDataKey` para criptografia de dados de saída. Para ler entradas criptografadas (como quando os dados de entrada já estão criptografados pelo Amazon S3), é necessária a permissão `kms:Decrypt`. O perfil do IAM precisa conceder essas permissões conforme necessário. No entanto, se a chave for de uma conta diferente da que está sendo usada atualmente, a política de chave do KMS para essa chave também deve conceder essas permissões ao perfil de acesso a dados da tarefa.

AWS políticas gerenciadas (predefinidas) para o Amazon Comprehend

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas AWS gerenciadas concedem as permissões necessárias para casos de uso comuns, para que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

As seguintes políticas AWS gerenciadas, que você pode anexar aos usuários em sua conta, são específicas do Amazon Comprehend:

- ComprehendFullAccess— Concede acesso total aos recursos do Amazon Comprehend, incluindo a execução de trabalhos de modelagem de tópicos. Inclui permissão para listar e obter perfis do IAM.
- ComprehendReadOnly— Concede permissão para executar todas as ações do Amazon Comprehend`StartEntitiesDetectionJob`, `StartDominantLanguageDetectionJob` exceto,,, `StartKeyPhrasesDetectionJob``StartSentimentDetectionJob`, e. `StartTargetedSentimentDetectionJob` `StartTopicsDetectionJob`

É preciso aplicar a seguinte política adicional a qualquer usuário que usará o Amazon Comprehend:

```
{
    "Version": "2012-10-17",
    "Statement":
      [
        {
          "Action":
            [
              "iam:PassRole"
            ],
          "Effect": "Allow",
          "Resource": "arn:aws:iam::*:role/*Comprehend*"
        }
      ]
  }
```

É possível revisar essas políticas de permissões gerenciadas fazendo login no console do IAM e pesquisando políticas específicas.

Essas políticas funcionam quando você está usando AWS SDKs ou a AWS CLI.

Você também pode criar suas próprias políticas do IAM personalizadas a fim de dar permissões para ações e recursos do Amazon Comprehend. Você pode anexar essas políticas personalizadas a usuários, perfis ou grupos que exijam essas permissões.

Permissões baseadas em perfis necessárias para operações assíncronas

Para usar as operações assíncronas do Amazon Comprehend, você deve conceder a ele o acesso ao bucket do Amazon S3 contendo sua coleção de documentos. Você faz isso criando um perfil de acesso a dados em sua conta com uma política de confiança para confiar na entidade principal do serviço do Amazon Comprehend. Para mais informações sobre como criar um perfil, consulte Criar um perfil para delegar permissões a um produto da AWS no Guia do usuário do Identity and Access Management da AWS .

O exemplo a seguir mostra um exemplo de política de confiança para o perfil criado. Para ajudar na prevenção do problema "confused deputy", você restringe o escopo da permissão usando uma ou mais chaves de contexto de condição global. Defina o valor `aws:SourceAccount` para o ID da sua conta. Se você usar a condição `ArnEquals`, defina o valor `aws:SourceArn` como o ARN da tarefa. Use um curinga para o número da tarefa no ARN, porque o Amazon Comprehend gera esse número como parte da criação de tarefas.

```
{
    "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/*"
          }
        }
      }
    ]
 }
```

Depois de criar o perfil, crie uma política de acesso para esse perfil. Isso deve conceder ao Amazon S3 as permissões `GetObject` e `ListBucket` para o bucket do Amazon S3 contendo seus dados de entrada, e a permissão de`PutObject` do Amazon S3 para o bucket de dados de saída do Amazon S3.

Conceder permissões a todas as ações do Amazon Comprehend

Depois de se inscrever AWS, você cria um usuário administrador para gerenciar sua conta, incluindo a criação de usuários e o gerenciamento de suas permissões.

Você pode optar por criar um usuário que tem permissões para todas as ações do Amazon Comprehend (pense nesse usuário como um administrador específico do serviço) para trabalhar com o Amazon Comprehend. Você pode anexar a política de permissões a seguir para este usuário.

```
{
  "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*"
    }
  ]
}
```

Essas permissões podem ser modificadas com relação à criptografia das seguintes maneiras:

- Para permitir que o Amazon Comprehend analise documentos armazenados em um bucket S3 criptografado, o perfil do IAM deve ter a permissão para `kms:Decrypt`.
- Para permitir que o Amazon Comprehend criptografe documentos armazenados em um volume de armazenamento anexado à instância de computação que processa a tarefa de análise, o perfil do IAM deve ter a permissão para `kms:CreateGrant`.
- Para permitir que o Amazon Comprehend criptografe os resultados de saída em seu bucket do S3, o perfil do IAM deve ter a permissão para `kms:GenerateDataKey`.

Permissões para ações de modelagem de tópicos

A política a seguir concede permissões ao usuário para realizar as operações de modelagem de tópicos do Amazon Comprehend.

```
{
  "Version": "2012-10-17",
  "Statement": [{
  "Sid": "AllowTopicModelingActions",
  "Effect": "Allow",
  "Action": [
            "comprehend:DescribeTopicsDetectionJob",
            "comprehend:ListTopicsDetectionJobs",
            "comprehend:StartTopicsDetectionJob",
         ],
        "Resource": "*"
        ]
    }
  ]
  }
```

Permissões necessárias para uma tarefa de análise assíncrona personalizada

> **Important**
>
> Se você tiver uma política do IAM que restringe o acesso ao modelo, não conseguirá concluir uma tarefa de inferência com um modelo personalizado. Sua política do IAM deve ser atualizada para ter um recurso curinga para uma tarefa personalizada de análise assíncrona.

Se você estiver usando o StartDocumentClassificationJob e StartEntitiesDetectionJob APIs, precisará atualizar sua política do IAM, a menos que esteja usando curingas como recursos no momento. Se você estiver StartEntitiesDetectionJobusando um modelo pré-treinado, isso não afetará você e você não precisará fazer nenhuma alteração.

O exemplo de política a seguir contém uma referência desatualizada.

```
{
    "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"
}
```

Essa é a política atualizada que você precisa usar para executar e. StartDocumentClassificationJob 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 políticas gerenciadas para o Amazon Comprehend

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas AWS gerenciadas do que escrever políticas você mesmo. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas AWS gerenciadas. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre políticas AWS gerenciadas, consulte políticas AWS gerenciadas no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas. Você não pode alterar as permissões nas políticas AWS gerenciadas. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem as permissões de uma política AWS gerenciada, portanto, as atualizações de políticas não violarão suas permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política ReadOnlyAccess AWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de perfis de trabalho, consulte Políticas gerenciadas pela AWS para perfis de trabalho no Guia do usuário do IAM.

AWS política gerenciada: ComprehendFullAccess

Essa política concede acesso total aos recursos do Amazon Comprehend, incluindo a execução de tarefas de modelagem de tópicos. Essa política também concede permissões de listagem e obtenção para buckets do Amazon S3 e perfis do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "comprehend:*",
                "iam:GetRole",
                "iam:ListRoles",
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
            ],
            "Resource": "*"
        }
    ]
}
```

AWS política gerenciada: ComprehendReadOnly

Essa política concede permissões de acesso somente para leitura para executar todas as ações do Amazon Comprehend, exceto as seguintes:

- `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": "*"
        }
    ]
}
```

Atualizações do Amazon Comprehend para políticas gerenciadas AWS

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amazon Comprehend desde que esse serviço começou a monitorar essas mudanças. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Histórico de documentos do Amazon Comprehend.

Alteração	Descrição	Data
ComprehendReadOnly: atualizar para uma política existente	O Amazon Comprehend agora permite as ações `comprehend:ListTargetedSentimentDetectionJobs` e ações `comprehend:DescribeTargetedSentimentDetectionJob` na política ComprehendReadOnly	30 de março de 2022
ComprehendReadOnly: atualizar para uma política existente	O Amazon Comprehend agora permite a ação `comprehend:DescribeResourcePolicy` na política ComprehendReadOnly	02 de fevereiro de 2022
ComprehendReadOnly: atualizar para uma política existente	O Amazon Comprehend agora permite as ações `ListEntityRecognizerSummaries` e ações `ListDocumentClassifierSummaries` na política ComprehendReadOnly	21 de setembro de 2021
ComprehendReadOnly: atualizar para uma política existente	O Amazon Comprehend agora permite a ação `ContainsPIIEntities` na política ComprehendReadOnly	26 de março de 2021

Alteração	Descrição	Data
O Amazon Comprehend começou a monitorar alterações	O Amazon Comprehend começou a monitorar as mudanças em suas políticas gerenciadas AWS .	1º de março de 2021

Solução de problemas de identidade e acesso do Amazon Comprehend

Use as seguintes informações para ajudar você a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o Amazon Comprehend e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amazon Comprehend
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas de fora da minha acessem meus Conta da AWS recursos do Amazon Comprehend

Não tenho autorização para executar uma ação no Amazon Comprehend

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para exibir detalhes sobre um recurso do *`my-example-widget`* fictício, mas não tem as permissões fictícias do `comprehend:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 comprehend:GetWidget on resource: my-example-widget
```

Nesse caso, a política de Mateo deve ser atualizada para permitir que ele tenha acesso ao recurso *`my-example-widget`* usando a ação `comprehend:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Caso receba uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, as políticas deverão ser atualizadas para permitir que você passe um perfil para o Amazon Comprehend.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro exemplificado a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon Comprehend. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha acessem meus Conta da AWS recursos do Amazon Comprehend

É possível criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon Comprehend é compatível com esses atributos, consulte Como o Amazon Comprehend funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Log de chamadas de API do Amazon Comprehend com o AWS CloudTrail

O Amazon Comprehend é integrado AWS CloudTrail com, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Comprehend. CloudTrail captura chamadas de API para o Amazon Comprehend como eventos. As chamadas capturadas incluem as chamadas do console do Amazon Comprehend e as chamadas de código para as operações da API do Amazon Comprehend. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amazon Comprehend. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Comprehend, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

Amazon Comprehend informações em CloudTrail

CloudTrail é ativado no seu Conta da AWS quando você cria a conta. Quando uma atividade de evento suportada ocorre no Amazon Comprehend, essa atividade é registrada em CloudTrail um evento junto com AWS outros eventos de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes no seu Conta da AWS. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para obter um registro contínuo dos eventos em sua empresa Conta da AWS, incluindo eventos do Amazon Comprehend, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a

todas as AWS regiões. A trilha registra eventos de todas as regiões na partição da AWS e entrega os arquivos de log no bucket do Amazon S3 que você especifica. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e Recebendo arquivos de CloudTrail log de várias contas

O Amazon Comprehend suporta o registro das seguintes ações como eventos CloudTrail em arquivos de log:

- 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

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário raiz.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Exemplo: entradas de arquivo de log do Amazon Comprehend

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `ClassifyDocument` ação.

```
{
```

```
    "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"
}
```

Validação de conformidade para o Amazon Comprehend

Os auditores externos avaliam a segurança e a conformidade do Amazon Comprehend como parte de vários programas de conformidade da AWS . Isso inclui PCI, FedRAMP, HIPAA e outros. Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios no AWS Artifact.

Sua responsabilidade relativa à compatibilidade quando for usado o Amazon Comprehend é determinada pela confidencialidade dos seus dados, pelos objetivos de compatibilidade da sua empresa e pelas leis e regulamentos aplicáveis. A AWS fornece os seguintes recursos para ajudar com a compatibilidade:

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados em segurança e conformidade em. AWS
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS Recursos de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Config— Esse AWS serviço avalia se suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte AWS Serviços no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

Resiliência no Amazon Comprehend

A infraestrutura AWS global é construída em torno de Região da AWS s e zonas de disponibilidade. Região da AWS s fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e altamente redundantes. Com as

zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre Região da AWS s e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança da infraestrutura no Amazon Comprehend

Como um serviço gerenciado, o Amazon Comprehend é protegido pela AWS segurança de rede global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Amazon Comprehend pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Diretrizes e cotas

A menos que especificado de outra forma, as cotas do Amazon Comprehend são por região. Você pode solicitar um aumento nas cotas ajustáveis, se necessário, para seus aplicativos. Para obter informações sobre as cotas e solicitar um aumento de cota, consulteAWS Service Quotas .

Tópicos

- Regiões do compatíveis
- Cotas para modelos integrados
- Cotas para modelos personalizados
- Cotas para flywheels

Regiões do compatíveis

O Amazon Comprehend está disponível nas seguintes regiões: AWS

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- AWS GovCloud (Oeste dos EUA)

Por padrão, o Amazon Comprehend fornece todas as operações de API em cada uma das Regiões compatíveis. Para ver as exceções, consulte Processamento de documentos.

Para obter informações sobre endpoints de API, consulte Regiões e endpoints do Amazon Comprehend na Referência geral da Amazon Web Services.

Para analisar as cotas atuais em uma região ou solicitar aumentos de cotas para cotas ajustáveis, abra o console do Service Quotas.

Cotas para modelos integrados

O Amazon Comprehend fornece modelos integrados para você analisar documentos de texto UTF-8. O Amazon Comprehend fornece operações síncronas e assíncronas que usam os modelos integrados.

Tópicos

- Análise em tempo real (síncrona)
- Análise assíncrona

Análise em tempo real (síncrona)

Esta seção descreve as cotas relacionadas a análises em tempo real usando os modelos integrados.

Tópicos

- Operações com um único documento
- Operações com vários documentos
- Solicitar controle de utilização para solicitações em tempo real (síncronas)

Operações com um único documento

A API do Amazon Comprehend fornece operações que usam um único documento como entrada. As cotas a seguir aplicam-se a essas operações.

Cotas gerais para operações com um único documento

As cotas a seguir se aplicam à análise em tempo real para detectar entidades, frases-chave ou Idioma dominante. Para detecção de entidades, essas cotas se aplicam à detecção com os modelos integrados. Para detecção de entidades personalizadas, consulte as cotas em Reconhecimento de entidade personalizado .

Descrição	Cota/diretriz
Tamanho máximo do documento	100 KB

Cotas específicas da operação para operações com um único documento

As cotas a seguir se aplicam à análise em tempo real para detectar sentimento, sentimento direcionado e sintaxe.

Descrição	Cota/diretriz
Tamanho máximo do documento	5 KB

Operações com vários documentos

A API do Amazon Comprehend fornece operações em lote que processam vários documentos com uma única solicitação de API. As cotas a seguir se aplicam às operações em lote.

Descrição	Cota/diretriz
Tamanho máximo do documento	5 KB
Máximo de documentos por solicitação	25

Para obter mais informações sobre o uso de operações com documentos em lote, consulte Processamento síncrono de vários documentos.

Solicitar controle de utilização para solicitações em tempo real (síncronas)

O Amazon Comprehend aplica controle de utilização às solicitações síncronas. Se a largura de banda de processamento do sistema estiver disponível, o Amazon Comprehend aumentará gradualmente o número de solicitações processadas. Para controlar o uso das operações síncronas da API pelo seu aplicativo, recomendamos que você ative os alertas de cobrança ou implemente a limitação de taxa em seu aplicativo.

Análise assíncrona

Esta seção descreve as cotas relacionadas à análise assíncrona usando os modelos integrados.

Cada operação assíncrona de API oferece suporte a um máximo de 10 trabalhos ativos. Para visualizar as cotas para cada operação de API, consulte a tabela de Service Quotas nos endpoints e cotas do Amazon Comprehend na Referência geral da Amazon Web Services.

É possível solicitar aumentos de cota para cotas ajustáveis do IAM usando o console do Service Quotas.

Tópicos

- Cotas gerais para operações assíncronas
- Cotas específicas da operação para trabalhos assíncronos
- Solicite controle de utilização para solicitações assíncronas

Cotas gerais para operações assíncronas

Você pode executar trabalhos de análise assíncronos usando o console ou qualquer uma das operações da API `Start*`. Para obter informações sobre quando usar operações assíncronas, consulte Processamento em lote assíncrono. As cotas a seguir se aplicam à maioria das operações de API `Start*` para modelos integrados. Para ver as exceções, consulte Cotas específicas da operação para trabalhos assíncronos.

Descrição	Cota/diretriz
Tamanho máximo de cada documento em trabalhos que detectam entidades, frases-chave, PII e idiomas	1 MB
Tamanho total máximo de todos os arquivos em uma solicitação	5 GB
Tamanho total mínimo de todos os arquivos em uma solicitação	500 bytes
Número máximo de arquivos, um documento por arquivo	1.000.000
Número total máximo de linhas, um documento por linha	1.000.000

Cotas específicas da operação para trabalhos assíncronos

Esta seção descreve as cotas para operações assíncronas específicas. Se uma cota não for especificada nas tabelas a seguir, o valor geral da cota se aplica.

Tópicos

- Sentimento
- Sentimento direcionado
- Eventos
- Modelagem de tópicos

Sentimento

Os trabalhos de sentimento assíncronos, que você cria com a StartSentimentDetectionJoboperação, têm as seguintes cotas.

Descrição	Cota/diretriz
Tamanho máximo de cada documento de entrada	5 KB

Sentimento direcionado

Os trabalhos assíncronos de sentimento direcionados, que você cria com a StartTargetedSentimentDetectionJoboperação, têm as seguintes cotas.

Descrição	Cota/diretriz
Formatos de documento compatíveis	UTF-8
Tamanho máximo de cada documento em um trabalho	10 KB
Tamanho máximo de todos os documentos em um trabalho	300 MB
Número máximo de arquivos, um documento por arquivo	30.000
Número total máximo de linhas, um documento por linha (para todos os arquivos solicitados)	30.000

Eventos

Os trabalhos de detecção de eventos assíncronos, que você cria com a StartEventsDetectionJoboperação, têm as seguintes cotas.

Descrição	Cotas
Codificação de caracteres	UTF-8
Tamanho total de todos os arquivos em um trabalho	50 MB
Tamanho máximo de cada documento em um trabalho	10 KB
Número máximo de arquivos, um documento por arquivo	5.000
Número máximo total de linhas, um documento por linha (para todos os arquivos solicitados)	5.000

Modelagem de tópicos

Os trabalhos assíncronos de modelagem de tópicos, que você cria com a StartTopicsDetectionJoboperação, têm as seguintes cotas.

Descrição	Cota/diretriz
Codificação de caracteres	UTF-8
O número máximo de tópicos a serem retornados	100
Tamanho máximo do arquivo para um arquivo, um documento por arquivo	100 MB

Para ter mais informações, consulte Modelagem de tópicos

Solicite controle de utilização para solicitações assíncronas

Cada operação assíncrona de API é compatível com um número máximo de solicitações por segundo (por região, por conta) e, também, com um máximo de 10 trabalhos ativos. Para visualizar

as cotas para cada operação de API, consulte a tabela de Service Quotas nos endpoints e cotas do Amazon Comprehend na Referência geral da Amazon Web Services.

É possível solicitar aumentos de cota para cotas ajustáveis do IAM usando o console do Service Quotas.

Cotas para modelos personalizados

Você pode usar o Amazon Comprehend para criar seus próprios modelos personalizados para classificação personalizada e reconhecimento personalizado de entidades. Esta seção fornece as diretrizes e cotas relacionadas ao treinamento e ao uso de modelos personalizados. Para obter mais informações sobre modelos personalizados, consulte Amazon Comprehend Custom.

Tópicos

- Cotas gerais
- Cotas para endpoints
- Classificação de documento
- Reconhecimento de entidade personalizado

Cotas gerais

O Amazon Comprehend define cotas de tamanho geral para cada tipo de documento de entrada que você pode analisar com modelos personalizados. Para cotas de análise em tempo real, consulte Tamanhos máximos de documentos para análise em tempo real. Para cotas de análise assíncrona, consulte Entradas para análise personalizada assíncrona.

Cada operação assíncrona de API é compatível com um número máximo de solicitações por segundo (por região, por conta) e, também, com um máximo de 10 trabalhos ativos. Para visualizar as cotas para cada operação de API, consulte a tabela de Service Quotas nos endpoints e cotas do Amazon Comprehend na Referência geral da Amazon Web Services.

É possível solicitar aumentos de cota para cotas ajustáveis do IAM usando o console do Service Quotas.

Cotas para endpoints

Você cria um endpoint para executar análises em tempo real com um modelo personalizado. Para obter informações sobre endpoints, consulte Gerenciar endpoints do Amazon Comprehend.

As cotas a seguir se aplicam aos endpoints. Para obter informações sobre como solicitar um aumento de cota, consulte AWS Service Quotas.

Descrição	Cota/diretriz
Número máximo de endpoints ativos por Região para cada conta	20
Número máximo de unidades de inferência por Região para cada conta	200
Número máximo de unidades de inferência por endpoint por região	50
Taxa de throughput por unidade de inferência (caracteres)	100/segundo
Taxa de throughput por unidade de inferência (documentos)	2/segundo

Classificação de documento

Esta seção descreve as diretrizes e cotas para as seguintes operações de classificação de documentos:

- Trabalhos de treinamento do classificador que você inicia com a CreateDocumentClassifieroperação.
- Tarefas assíncronas de classificação de documentos que você inicia com a operação. StartDocumentClassificationJob
- Solicitações síncronas de classificação de documentos que usam a ClassifyDocumentoperação.

Cotas gerais para classificação de documentos

A tabela a seguir descreve as cotas gerais relacionadas ao treinamento de classificadores personalizados.

Descrição	Cota/diretriz
Comprimento máximo do nome da classe	5.000 caracteres

Descrição	Cota/diretriz
Número de classes (modo multiclasse)	2 a 1.000
Número de classes (modo multirrótulo)	2 a 100
Formato de anotações	
Número mínimo de anotações por classe (modo multiclasse)	10
Número mínimo de anotações por classe (modo multirrótulo)	10
Número mínimo de anotações (modo multirrótulo)	50
Formato de arquivo CSV	
Número mínimo de documentos de treinamento por classe (modo multiclasse)	50
Número mínimo de documentos de treinamento por classe (modo multirrótulo)	10
Número mínimo de documentos de treinamento (modo multirrótulo)	50

Classificação para documentos de texto simples

Você cria e treina um modelo de texto simples usando documentos de entrada de texto simples. O Amazon Comprehend fornece operações assíncronas e em tempo real para classificar documentos de texto simples usando um modelo de texto simples.

Treinamento

A tabela a seguir descreve as cotas relacionadas ao treinamento de um classificador personalizado com documentos de texto simples.

Descrição	Cota/diretriz
Tamanho total de todos os arquivos no trabalho de treinamento	5 GB

Descrição	Cota/diretriz
Número máximo de arquivos manifesto aumentados para treinar um classificador personalizado	5
Número máximo de nomes de atributos para cada arquivo manifesto aumentado	5
Comprimento máximo do nome do atributo	63 caracteres

Análise em tempo real (síncrona)

A tabela a seguir descreve as cotas relacionadas à classificação em tempo real de documentos de texto simples.

Descrição	Cota/diretriz
Número máximo de documentos por solicitação síncrona	1
Tamanho máximo do documento de texto (codificação UTF-8)	10 KB

Análise assíncrona

A tabela a seguir descreve as cotas relacionadas à classificação assíncrona de documentos de texto simples.

Descrição	Cota/diretriz
Tamanho total de todos os arquivos no trabalho assíncrono	5 GB
Tamanho máximo do arquivo para um arquivo, um documento por arquivo	10 MB
Número máximo de arquivos, um documento por arquivo	1.000.000
Número máximo total de linhas, um documento por linha (para todos os arquivos solicitados)	1.000.000

Classificação para documentos semiestruturados

Esta seção descreve as diretrizes e cotas para classificação de documentos semiestruturados. Para classificar documentos semiestruturados, use um modelo de documento nativo que você treinou com documentos de entrada nativos.

Treinando um modelo de documento nativo com documentos semiestruturados

A tabela a seguir descreve as cotas relacionadas ao treinamento de um classificador personalizado com documentos semiestruturados, como documentos em PDF, documentos do Word e arquivos de imagem.

Descrição	Cota/diretriz
Número máximo de páginas em todos os documentos	10.000
Tamanho máximo do arquivo de anotações (todos os tamanhos de CSV combinados)	5 MB
Tamanho do corpus do documento (documentos de treinamento e de teste)	10 GB
Tamanhos de arquivo para treinamento e teste de arquivos	
Tamanho do arquivo de imagem (JPG, PNG, TIFF).	1 byte a 10 MB. Arquivos TIFF: no máximo uma página.
Tamanho da página para documentos em PDF	1 byte a 10 MB
Tamanho da página para documentos do Word	1 byte a 10 MB
Tamanho do JSON de saída da API do Amazon Textract	1 byte a 1 MB

Análise em tempo real (síncrona)

Esta seção descreve as cotas relacionadas à classificação em tempo real de documentos semiestruturados.

A tabela a seguir mostra os tamanhos máximos de arquivo dos documentos de entrada. Para todos os tipos de documentos de entrada, o máximo do arquivo de entrada é de uma página, com no máximo 10.000 caracteres.

Tipo de arquivo	Tamanho máximo (API)	Tamanho máximo (console)
Documentos de texto em UTF-8	10 KB	10 KB
Documentos em PDF	10 MB	5 MB
Documentos do Word	10 MB	5 MB
Arquivos de imagem	10 MB	5 MB
Tamanho da saída da API do Amazon Textract	1 MB	n/a

Análise assíncrona

A tabela a seguir descreve as cotas relacionadas à classificação assíncrona de documentos semiestruturados.

Descrição	Cota/diretriz
Número máximo de páginas em todos os documentos de entrada para um trabalho	25.000
Tamanho do corpus do documento	25 GB
Tamanho do arquivo de imagem (JPG, PNG ou TIFF)	1 byte a 10 MB. Arquivos TIFF: no máximo uma página.
Tamanho da página para documentos em PDF	1 byte a 10 MB
Tamanho da página para documentos do Word	1 byte a 10 MB
Tamanho do JSON de saída da API do Textract	1 byte a 1 MB.

Reconhecimento de entidade personalizado

Esta seção descreve as diretrizes e cotas para as seguintes operações de reconhecimento de entidades personalizado:

- Os trabalhos de treinamento do reconhecedor de entidades começaram com a CreateEntityRecognizeroperação.
- Os trabalhos assíncronos de reconhecimento de entidades foram iniciados com a operação. StartEntitiesDetectionJob
- Solicitações síncronas de reconhecimento de entidades usando a DetectEntitiesoperação.

Reconhecimento de entidades personalizado para documentos de texto simples

O Amazon Comprehend fornece operações assíncronas e síncronas para analisar documentos de texto simples com um reconhecedor de entidades personalizado.

Treinamento

Esta seção descreve as cotas relacionadas ao treinamento de um reconhecedor de entidades personalizado para analisar documentos de texto simples. Para treinar o modelo, você pode fornecer uma lista de entidades ou um conjunto de documentos de texto anotados.

A tabela a seguir descreve as cotas relacionadas ao treinamento do modelo com uma lista de entidades.

Descrição	Cota/diretriz
Número de entidades por modelo	1 a 25
Tamanho do documento (UTF-8)	1 a 5.000 bytes
Número de itens na lista de entidades	1 a 1 milhão
Comprimento da entrada individual (pós-remoção) na lista de entradas	1 a 5.000
Tamanho da lista de entradas (todos os documentos em texto simples combinados)	5 KB a 200 MB

A tabela a seguir descreve as cotas relacionadas ao treinamento do modelo com documentos de texto anotados.

Descrição	Cota/diretriz
Número de entidades por modelo/reconhecedor de entidade personalizado	1 a 25
Tamanho do documento (UTF-8)	1 a 5.000 bytes
Número de documentos (consulte Anotações em texto simples)	3 a 200.000
Tamanho do corpo do documento (todos os documentos em texto simples combinados)	5 KB a 200 MB
Número mínimo de anotações por entidade	25

Análise em tempo real (síncrona)

A tabela a seguir descreve as cotas relacionadas à análise em tempo real de documentos de texto sem formatação.

Descrição	Cota/diretriz
Número máximo de documentos por solicitação síncrona	1
Tamanho máximo do documento de texto (codificação UTF-8)	5 KB

Análise assíncrona

A tabela a seguir descreve as cotas relacionadas ao reconhecimento assíncrono de entidades de documentos de texto simples.

Descrição	Cota/diretriz
Tamanho do documento (UTF-8)	1 byte a 1 MB
Número máximo de arquivos, um documento por arquivo	1.000.000

Descrição	Cota/diretriz
Número máximo total de linhas, um documento por linha (para todos os arquivos solicitados)	1.000.000
Tamanho do corpo do documento (todos os documentos em texto simples combinados)	1 byte a 5 GB

Reconhecimento de entidades personalizado para documentos semiestruturados

O Amazon Comprehend fornece operações assíncronas e síncronas para analisar documentos semiestruturados com um reconhecedor de entidades personalizado. Você deve treinar o modelo usando documentos em PDF anotados.

Treinamento

A tabela a seguir descreve as cotas relacionadas ao treinamento de um reconhecedor de entidades personalizado (CreateEntityRecognizer) para analisar documentos semiestruturados.

Descrição	Cota/diretriz
Número de entidades por modelo/reconhecedor de entidade personalizado	1 a 25
Tamanho máximo do arquivo de anotação (UTF-8 JSON)	5 MB
Número de documentos	250 a 10.000
Tamanho do corpo do documento (todos os documentos em texto simples combinados)	5 KB a 1 GB
Número mínimo de anotações por entidade	100
Número máximo de arquivos manifesto aumentados para treinar um reconhecedor de entidades personalizado	5
Número máximo de nomes de atributos para cada arquivo manifesto aumentado	5

Descrição	Cota/diretriz
Comprimento máximo do nome do atributo	63 caracteres

Análise em tempo real (síncrona)

Esta seção descreve as cotas relacionadas à análise em tempo real de documentos semiestruturados.

A tabela a seguir mostra os tamanhos máximos de arquivo dos documentos de entrada. Para todos os tipos de documentos de entrada, o máximo do arquivo de entrada é de uma página, com no máximo 10.000 caracteres.

Tipo de arquivo	Tamanho máximo (API)	Tamanho máximo (console)
Documentos de texto em UTF-8	10 KB	10 KB
Documentos em PDF	10 MB	5 MB
Documentos do Word	10 MB	5 MB
Arquivos de imagem	10 MB	5 MB
Arquivos de saída do Textract	1 MB	n/a

Análise assíncrona

Esta seção descreve as cotas para análise assíncrona de documentos semiestruturados.

Descrição	Cota/diretriz
Tamanho da imagem (JPG ou PNG)	1 byte a 10 MB
Tamanho da imagem (TIFF)	1 byte a 10 MB. Máximo de uma página.
Tamanho do documento (PDF)	1 byte a 50 MB

Descrição	Cota/diretriz
Tamanho do documento (Docx)	1 byte a 5 MB
Tamanho do documento (UTF-8)	1 byte a 1 MB
Número máximo de arquivos, um documento por arquivo (um documento por linha não é permitido para arquivos de imagem ou documentos em PDF/do Word)	500
Número máximo de páginas para um arquivo PDF ou Docx	100
Tamanho do corpus do documento após a extração do texto (texto simples, todos os arquivos combinados)	1 byte a 5 GB

Para obter mais informações sobre limites para imagens, consulte Limites rígidos no Amazon Textract

Cotas para flywheels

Use flywheels para gerenciar o treinamento e o monitoramento de versões de modelos personalizados para classificação personalizada e reconhecimento personalizado de entidades. Para obter mais informações sobre flywheels, consulte Flywheels.

Cotas gerais para flywheels

As cotas a seguir se aplicam flywheels e às iterações de flywheel.

Descrição	Cota/diretriz
Número máximo de flywheels	50
Número máximo de flywheels no status CREATING	10
Número máximo de conjuntos de dados de treinamento por flywheel	50
Número máximo de conjuntos de dados de teste por flywheel	50

Descrição	Cota/diretriz
Número máximo de conjuntos de dados com status INGESTING	10
Número máximo de iterações de flywheel em andamento por conta	10

Cotas de conjuntos de dados para modelos de classificação personalizados

Quando você ingere um conjunto de dados para um flywheel associado a um modelo de classificação personalizado, as cotas a seguir se aplicam.

Descrição	Cota/diretriz
Número mínimo de documentos de treinamento por classe (modo multirrótulo)	50
Número máximo de documentos de treinamento	1.000.000
Tamanho mínimo do conjunto de dados	500 bytes
Tamanho máximo do conjunto de dados	5 GB
Tamanho máximo do arquivo para um arquivo, um documento por arquivo	10 MB

Cotas de conjuntos de dados para modelos personalizados de reconhecimento de entidades

Quando você ingere um conjunto de dados para um flywheel associado a um modelo de reconhecimento de entidade personalizado, as cotas a seguir se aplicam.

Descrição	Cota/diretriz
Tamanho máximo do documento	5 KB
Número mínimo de documentos de treinamento	3

Descrição	Cota/diretriz
Número máximo de documentos de treinamento	200.000
Número mínimo de anotações por entidade	25
Tamanho máximo do conjunto de dados	200 MB

Tutoriais e outros recursos

Tutoriais e outros recursos para o Amazon Comprehend.

Tópicos

- Tutorial: analisando insights de avaliações de clientes com o Amazon Comprehend
- Utilizar pontos de acesso do Amazon S3 object Lambda para informações de identificação pessoal (PII)
- Solução: analisar texto com o Amazon Comprehend e OpenSearch

Tutorial: analisando insights de avaliações de clientes com o Amazon Comprehend

Este tutorial explica como usar o Amazon Comprehend com o Amazon Simple Storage AWS GlueService Amazon Athena,, e a QuickSight Amazon para obter informações valiosas sobre seus documentos. O Amazon Comprehend pode extrair sentimentos (o “clima” de um documento) e entidades (nomes de pessoas, organizações, eventos, datas, produtos, lugares, quantidades e títulos) a partir de textos não estruturados.

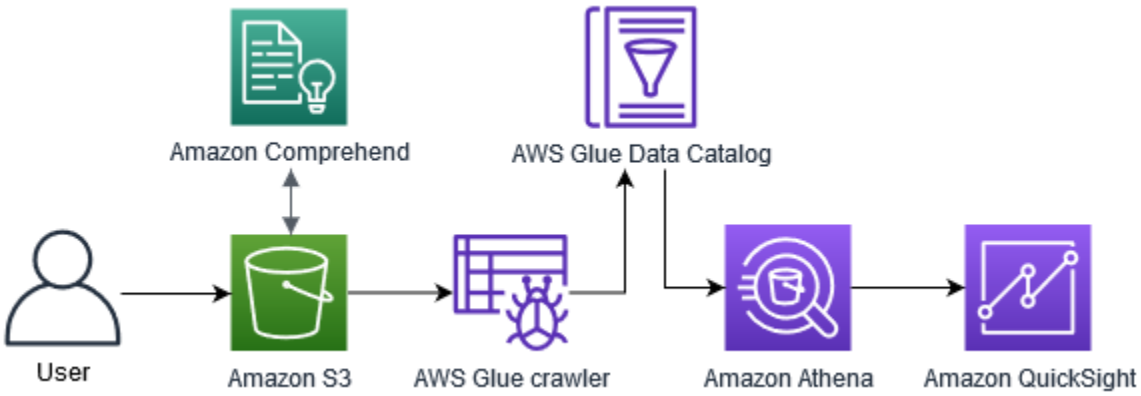
Por exemplo, você pode obter informações úteis a partir de avaliações de clientes. Neste tutorial, você analisa um conjunto de dados de amostras de avaliações de clientes sobre um livro. Você usa a análise de sentimentos do Amazon Comprehend para determinar se os clientes são positivos ou negativos em relação ao livro. Você também usa a análise de entidades do Amazon Comprehend para descobrir menções a entidades importantes, como livros ou autores relacionados. Depois de seguir este tutorial, você poderá descobrir que mais de 50% das avaliações são positivas. Você também pode descobrir que os clientes estão comparando autores e expressando interesse em outros livros clássicos.

Neste tutorial, você conseguirá fazer o seguinte:

- Armazenar um conjunto de dados de amostras no Amazon Simple Storage Service (Amazon S3). O Amazon Simple Storage Service é um serviço de armazenamento de objetos.
- Use o Amazon Comprehend para analisar o sentimento e as entidades nos documentos de revisão.

- Use um AWS Gluerastreador para armazenar os resultados da análise em um banco de dados. AWS Glue é um serviço de extração, transformação e carregamento (ETL) que permite catalogar e limpar seus dados para análise.
- Execute Amazon Athenaconsultas para limpar seus dados. Amazon Athena é um serviço de consulta interativo sem servidor.
- Crie visualizações com seus dados na Amazon QuickSight. QuickSight é uma ferramenta de inteligência de negócios sem servidor para extrair insights de seus dados.

O diagrama mostra o fluxo de trabalho.



Tempo estimado para concluir este tutorial: 1 hora

Custo estimado: algumas das ações deste tutorial geram cobranças na sua Conta da AWS. Para obter informações sobre as cobranças de cada um desses serviços, consulte as páginas de preços a seguir.

- Preços do Amazon S3
- Preços do Amazon Comprehend
- AWS Glue preços
- Definição de preços do Amazon Athena
- Definição de preços do QuickSight

Tópicos

- Pré-requisitos
- Etapa 1: adicionando documentos ao Amazon S3
- Etapa 2: (somente para CLI) criando um perfil do IAM para o Amazon Comprehend

- Etapa 3: executando trabalhos de análise em documentos no Amazon S3
- Etapa 4: preparar a saída do Amazon Comprehend para visualização de dados
- Etapa 5: visualizando o resultado do Amazon Comprehend em QuickSight

Pré-requisitos

Para concluir este tutorial, você precisará do seguinte:

- Um Conta da AWS. Para obter informações sobre como configurar um Conta da AWS, consulteConfiguração.
- Uma entidade do IAM (usuário, grupo ou perfil). Para saber como configurar usuários e grupos para sua conta, consulte o tutorial de Conceitos básicos no Guia do usuário do IAM.
- As política de permissões a seguir a um usuário, grupo ou perfil. A política concede algumas das permissões necessárias para concluir este tutorial. O próximo pré-requisito descreve as permissões adicionais de que você precisa.

```
{
  "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*"
      ]
    }
  ]
}
```

Use a política anterior para criar uma política do IAM e a anexe ao grupo ou usuário. Para obter mais informações sobre como criar uma política do IAM, consulte Criar políticas do IAM no Guia do usuário do IAM. Para obter informações sobre como anexar a política do IAM, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

- Políticas gerenciadas anexadas a um grupo ou usuário do IAM. Além da política anterior, você também deve anexar as seguintes políticas AWS gerenciadas ao seu grupo ou usuário:
  - `AWSGlueConsoleFullAccess`
  - `AWSQuicksightAthenaAccess`

Essas políticas gerenciadas dão permissão para usar AWS Glue Amazon Athena, QuickSight e. Para obter informações sobre como anexar a política do IAM, consulte Adicionar e remover permissões de identidade do IAM no Guia do usuário do IAM.

Etapa 1: adicionando documentos ao Amazon S3

Antes de iniciar os trabalhos de análise do Amazon Comprehend, você precisa armazenar um conjunto de dados de amostra de avaliações de clientes no Amazon Simple Storage Service (Amazon S3). O Amazon S3 hospeda seus dados em contêineres chamados buckets. O Amazon Comprehend pode analisar documentos armazenados em um bucket e enviar os resultados da análise para um bucket. Nesta etapa, você cria um bucket do S3, cria pastas de entrada e saída no bucket e carrega um conjunto de dados de amostra para o bucket.

Tópicos

- Pré-requisitos
- Baixe os dados de amostra
- Criar um bucket do Amazon S3
- (Somente console) crie pastas
- Carregue os dados de entrada

Pré-requisitos

Antes de começar, revise o Tutorial: analisando insights de avaliações de clientes com o Amazon Comprehend e conclua os pré-requisitos.

Baixe os dados de amostra

O conjunto de dados de amostra a seguir contém avaliações da Amazon retiradas do conjunto de dados maior "Amazon reviews - Full", que foi publicado com o artigo "Character-level Convolutional Networks for Text Classification" (Xiang Zhang et al., 2015). Baixe o conjunto de dados para o computador.

Obter os dados de amostra

1. Baixe o arquivo zip tutorial-reviews-data.zip para o seu computador.

2. Extraia o arquivo .zip em seu computador local. Há dois arquivos. O arquivo `THIRD_PARTY_LICENSES.txt` é a licença de código aberto para o conjunto de dados publicado por Xiang Zhang et al. O arquivo `amazon-reviews.csv` é o conjunto de dados que você analisa no tutorial.

Criar um bucket do Amazon S3

Depois de baixar o conjunto de dados de amostra, crie um bucket do Amazon S3 para armazenar seus dados de entrada e saída. Você pode criar um bucket do S3 usando o console do Amazon S3 ou o AWS Command Line Interface (AWS CLI).

Crie um bucket do Amazon S3 (console)

No console do Amazon S3, você cria um bucket com um nome exclusivo em todos os AWS.

Para criar um bucket do S3 (console)

1. Faça login no AWS Management Console e abra o console do Amazon S3 em. https://console.aws.amazon.com/s3/
2. Em Buckets, escolha Criar bucket.
3. Em Nome do bucket, insira um nome globalmente exclusivo que descreva a finalidade do bucket.
4. Em Região, escolha a AWS região em que você deseja criar o bucket. A região que você escolher deve oferecer suporte ao Amazon Comprehend. Para reduzir a latência, escolha a AWS região mais próxima da sua localização geográfica que seja suportada pelo Amazon Comprehend. Para obter uma lista das regiões que são compatíveis com o Amazon Comprehend, consulte a Tabela de regiões no Guia global de infraestrutura.
5. Deixe as configurações padrão para Propriedade do objeto, Configurações de bucket para bloquear acesso público, Versionamento de bucket e Tags.
6. Em Criptografia padrão, escolha Desabilitar.

> **Tip**
>
> Embora este tutorial não use criptografia, talvez você queira usar criptografia ao analisar dados importantes. Para end-to-end criptografia, você pode criptografar seus dados em repouso no bucket e também ao executar trabalhos de análise. Para obter mais

informações sobre criptografia com AWS, consulte O que é AWS Key Management Service? no Guia do AWS Key Management Service desenvolvedor.

7. Revise as configurações do bucket e escolha Criar bucket.

Crie um bucket do Amazon S3 (AWS CLI)

Depois de abrir o AWS CLI, você executa o `create-bucket` comando para criar um bucket que armazenará os dados de entrada e saída.

Como criar um bucket do Amazon S3 (AWS CLI)

1. Para criar seu bucket, execute o comando a seguir no AWS CLI. Substitua amzn-s3-demo-bucket por um nome para o bucket que seja exclusivo em todos os. AWS

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket
```

Por padrão, o `create-bucket` comando cria um bucket na `us-east-1` AWS região. Para criar um bucket em uma Região da AWS diferente de `us-east-1`, adicione o parâmetro `LocationConstraint` para especificar sua Região. Por exemplo, o comando a seguir cria um bucket na Região `us-west-2`.

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket
--region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2
```

Observe que somente algumas regiões oferecem suporte ao Amazon Comprehend. Para obter uma lista das regiões que são compatíveis com o Amazon Comprehend, consulte a Tabela de regiões no Guia global de infraestrutura.

2. Para garantir que o bucket foi criado com êxito, execute o comando a seguir. O comando lista todos os buckets do S3 associados à sua conta.

```
aws s3 ls
```

(Somente console) crie pastas

Depois, crie duas pastas no seu bucket do S3. A primeira pasta é para seus dados de entrada. A segunda pasta é para onde o Amazon Comprehend envia os resultados da análise. Se você usa o

console do Amazon S3, você precisa criar as pastas manualmente. Se você usar o AWS CLI, poderá criar pastas ao carregar o conjunto de dados de amostra ou executar um trabalho de análise. Por esse motivo, fornecemos um procedimento para criar pastas somente para usuários do console. Se você estiver usando o AWS CLI, você criará pastas dentro de Carregue os dados de entrada e dentro de Etapa 3: executando trabalhos de análise em documentos no Amazon S3.

Para criar pastas em seu bucket do S3 (console)

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Em Buckets, escolha seu bucket na lista de buckets.
3. Na guia Visão geral, escolha Criar pasta.
4. Para o novo nome da pasta, insira `input`.
5. Para as configurações de criptografia, escolha (Nenhuma (use as configurações do bucket).
6. Escolha Salvar.
7. Repita as etapas de 3 a 6 para criar outra pasta para a saída dos trabalhos de análise, mas na etapa 4, insira o `output` do nome da nova pasta.

Carregue os dados de entrada

Agora que você tem um bucket, carregue o conjunto de dados de amostra `amazon-reviews.csv`. Você pode carregar dados para buckets do S3 com o console Amazon S3 ou o AWS CLI.

Carregue documentos de amostra para um bucket (console)

No console do Amazon S3, carregue o arquivo do conjunto de dados de amostra para a pasta de entrada.

Par carregar os documentos de amostra (console)

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Em Buckets, escolha seu bucket na lista de buckets.
3. Escolha a pasta `input` e, em seguida, escolha Upload.
4. Escolha Adicionar arquivos e, em seguida, escolha o arquivo `amazon-reviews.csv` no seu computador.
5. Deixe as outras configurações nos valores padrão.
6. Escolha Carregar.

Carregue os documentos de amostra em um bucket (AWS CLI)

Crie uma pasta de entrada em seu bucket do S3 e carregue o arquivo do conjunto de dados na nova pasta com o comando `cp`.

Par carregar os documentos de amostra (AWS CLI)

1. Para fazer o upload do `amazon-reviews.csv` arquivo em uma nova pasta no seu bucket, execute o AWS CLI comando a seguir. Substitua amzn-s3-demo-bucket pelo nome do seu bucket. Ao adicionar o caminho do `/input/` no final, o Amazon S3 cria automaticamente uma nova pasta chamada `input` em seu bucket e carrega o arquivo do conjunto de dados nessa pasta.

   ```
   aws s3 cp amazon-reviews.csv s3://amzn-s3-demo-bucket/input/
   ```

2. Para garantir que o arquivo foi carregado com êxito, execute o comando a seguir. O comando lista o conteúdo da pasta `input` do seu bucket.

   ```
   aws s3 ls s3://amzn-s3-demo-bucket/input/
   ```

Agora, você tem um bucket do S3 com o arquivo `amazon-reviews.csv` em uma pasta chamada `input`. Se você usou o console, você também tem uma pasta `output` no bucket. Se você usou o AWS CLI, você criará a pasta de saída ao executar os trabalhos de análise do Amazon Comprehend.

Etapa 2: (somente para CLI) criando um perfil do IAM para o Amazon Comprehend

Essa etapa é necessária somente se você estiver usando o AWS Command Line Interface (AWS CLI) para concluir este tutorial. Se você estiver usando o console do Amazon Comprehend para executar os trabalhos de análise, vá para Etapa 3: executando trabalhos de análise em documentos no Amazon S3.

Para executar trabalhos de análise, o Amazon Comprehend exige acesso ao bucket do Amazon S3 que contém o conjunto de dados de amostra e conterá a saída dos trabalhos. As funções do IAM permitem que você controle as permissões de AWS serviços ou usuários. Nesta etapa, você criará um perfil do IAM para o Amazon Comprehend. Em seguida, você cria e anexa a esse perfil uma política baseada em recursos que concede ao Amazon Comprehend acesso ao seu bucket do S3.

Ao final dessa etapa, o Amazon Comprehend terá as permissões necessárias para acessar seus dados de entrada, armazenar sua saída e executar trabalhos de análise de sentimentos e entidades.

Para obter mais informações sobre como usar o IAM com o Amazon Comprehend, consulte Como o Amazon Comprehend funciona com o IAM.

Tópicos

- Pré-requisitos
- Criar um perfil do IAM
- Anexe a política do IAM ao perfil do IAM.

Pré-requisitos

Antes de começar, faça o seguinte:

- Completa Etapa 1: adicionando documentos ao Amazon S3.
- Tenha um editor de código ou texto para salvar políticas JSON e acompanhar seus nomes de recursos da Amazon (ARNs).

Criar um perfil do IAM

Para acessar seu bucket do Amazon Simple Storage Service (Amazon S3), o Amazon Comprehend precisa assumir uma função (IAM). AWS Identity and Access Management O perfil do IAM declara o Amazon Comprehend como uma entidade confiável. Depois que o Amazon Comprehend assumir o perfil e se tornar uma entidade confiável, você poderá conceder ao bucket permissões de acesso ao Amazon Comprehend. Nesta etapa, você cria um perfil que rotula o Amazon Comprehend como uma entidade confiável. Você pode criar uma função com o console do Amazon Comprehend AWS CLI ou com o console do Amazon Comprehend. Para usar o console, vá para Etapa 3: executando trabalhos de análise em documentos no Amazon S3.

O console do Amazon Comprehend permite que você selecione perfis em que o nome do perfil contenha 'Comprehend' e em que a política de confiança inclua comprehend.amazonaws.com. Configure suas funções criadas pela CLI para atender a esses critérios se você quiser que o console as exiba.

Para criar uma função do IAM para o Amazon Comprehend AWS (CLI)

1. Salve a política de confiança a seguir como um documento JSON chamado `comprehend-trust-policy.json` em um editor de códigos ou texto em seu computador. Essa política de confiança declara o Amazon Comprehend como uma entidade confiável e permite que ele assuma um perfil do IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Para criar a função do IAM, execute o AWS CLI comando a seguir. O comando cria um perfil do IAM chamado `AmazonComprehendServiceRole-access-role` e anexa a política de confiança a ele. Substitua *`path/`* pelo caminho do seu computador local até o documento JSON.

```
aws iam create-role --role-name AmazonComprehendServiceRole-access-role
--assume-role-policy-document file://path/comprehend-trust-policy.json
```

> **Tip**
>
> Se você receber uma mensagem de Erro ao analisar o parâmetro, o caminho para seu arquivo de política de confiança JSON provavelmente está incorreto. Forneça o caminho relativo para o arquivo com base no seu diretório inicial.

3. Copie o nome do recurso da Amazon (ARN) e salve-o em um editor de texto. O ARN tem um formato semelhante a *`arn:aws:iam::123456789012:role/AmazonComprehendServiceRole-access-role`*. Você precisa desse ARN para executar trabalhos de análise do Amazon Comprehend.

Anexe a política do IAM ao perfil do IAM.

Para acessar seu bucket do Amazon S3, o Amazon Comprehend precisa de permissões para listar, ler e gravar. Para dar ao Amazon Comprehend as permissões necessárias, crie e anexe uma política do IAM para um perfil do IAM. A política do IAM permite que o Amazon Comprehend recupere os dados de entrada do seu bucket e grave os resultados da análise no bucket. Depois de criar a política, anexe-a ao seu perfil do IAM.

Para criar uma política do IAM (AWS CLI)

1. Salve a seguinte política localmente como um documento JSON chamado `comprehend-access-policy.json`. Ela concede ao Amazon Comprehend acesso ao bucket do S3 especificado.

```
{
    "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. Para criar a política de acesso ao bucket do S3, execute o AWS CLI comando a seguir. Substitua *path/* pelo caminho do seu computador local até o documento JSON.

```
aws iam create-policy --policy-name comprehend-access-policy
--policy-document file://path/comprehend-access-policy.json
```

3. Copie o ARN da política de acesso e salve-o em um editor de texto. O ARN tem um formato semelhante a *arn:aws:iam::123456789012:policy/comprehend-access-policy*. Você precisa desse ARN para vincular sua política de acesso ao seu perfil do IAM.

Para anexar a política do IAM ao perfil do IAM (AWS CLI)

- Execute o seguinte comando: Substitua *policy-arn* pelo ARN da política de acesso que você copiou na etapa anterior.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name AmazonComprehendServiceRole-access-role
```

Agora você tem um perfil do IAM chamado `AmazonComprehendServiceRole-access-role` que tem uma política de confiança para o Amazon Comprehend e uma política de acesso que concede ao Amazon Comprehend acesso ao seu bucket do S3. O ARN do perfil do IAM também será copiado para um editor de texto.

Etapa 3: executando trabalhos de análise em documentos no Amazon S3

Depois de armazenar os dados no Amazon S3, você pode começar a executar trabalhos de análise no Amazon Comprehend. Um trabalho de análise de sentimentos determina o clima geral de um documento (positivo, negativo, neutro ou misto). Um trabalho de análise de entidades extrai os nomes de objetos do mundo real de um documento. Esses objetos incluem pessoas, lugares, títulos, eventos, datas, quantidades, produtos e organizações. Nesta etapa, você executa dois trabalhos de análise do Amazon Comprehend para extrair o sentimento e as entidades do conjunto de dados de amostra.

Tópicos

- Pré-requisitos
- Analise sentimentos e entidades

Pré-requisitos

Antes de começar, faça o seguinte:

- Completa Etapa 1: adicionando documentos ao Amazon S3.
- (Opcional) Se você estiver usando o AWS CLI, preencha Etapa 2: (somente para CLI) criando um perfil do IAM para o Amazon Comprehend e tenha seu ARN de função do IAM pronto.

Analise sentimentos e entidades

O primeiro trabalho que você executa analisa o sentimento de cada avaliação do cliente no conjunto de dados de amostra. O segundo trabalho extrai as entidades em cada avaliação do cliente. Você pode realizar trabalhos de análise do Amazon Comprehend usando o console do Amazon Comprehend ou o AWS CLI.

> **Tip**
>
> Verifique se você está em uma AWS região que ofereça suporte ao Amazon Comprehend. Para obter mais informações, consulte a Tabela de regiões no Guia Global de Infraestrutura.

Analise sentimentos e entidades (console)

Ao usar o console do Amazon Comprehend, você cria um trabalho por vez. Você precisa repetir as etapas a seguir para executar um trabalho de análise de sentimentos e de entidades. Observe que, para o primeiro trabalho, você cria um perfil do IAM, mas para o segundo trabalho, você pode reutilizar o perfil do IAM do primeiro trabalho. Você pode reutilizar o perfil do IAM, desde que use o mesmo bucket e pastas do S3.

Para executar trabalhos de análise de sentimentos e entidades (console)

1. Certifique-se de que você esteja na mesma Região em que você criou o bucket do Amazon Simple Storage Service (Amazon S3). Se você estiver em outra região, na barra de navegação, escolha a AWS região em que você criou seu bucket do S3 no seletor de regiões.

2. Abra o console do Amazon Comprehend em https://console.aws.amazon.com/comprehend/
3. Escolha Executar o Amazon Comprehend).
4. No painel de navegação, escolha Trabalhos de análise.
5. Escolha Criar trabalho.
6. Na seção Configurações de trabalho, faça o seguinte:
    a. Em Nome, insira `reviews-sentiment-analysis`.
    b. Em Tipo de análise, escolha Sentiment.
    c. Em Idioma, escolha Inglês.
    d. Deixe a configuração de Criptografia do trabalho desabilitada.
7. Na seção Dados de entrada, faça o seguinte:
    a. Em Fonte de dados, escolha Meus documentos.
    b. Para a Local do S3, escolha Procurar no S3 e, em seguida, escolha seu bucket na lista de buckets.
    c. Em seu bucket do S3, em Objetos, escolha sua pasta `input`.
    d. Na pasta `input`, escolha o conjunto de dados de amostra `amazon-reviews.csv` e, em seguida, Escolher.
    e. Em Formato de entrada, escolha Um documento por linha.
8. Na seção Dados de saída, faça o seguinte:
    a. Para a Local do S3, escolha Procurar no S3 e, em seguida, escolha seu bucket na lista de buckets.
    b. Em seu bucket do S3, em Objetos, escolha a pasta `output` e, em seguida, Escolher.
    c. Deixe a Criptografia desabilitada.
9. Na seção Permissões de acesso, faça o seguinte:
    a. Em Perfil do IAM), escolha Criar um perfil do IAM).
    b. Em Permissões para acessar, escolha Buckets do S3 de entrada e saída.
    c. Em Sufixo do nome, insira `comprehend-access-role`. Esse perfil fornece acesso ao bucket do Amazon S3.
10. Escolha Criar trabalho.
11. Repita as etapas de 1 a 10 para criar um trabalho de análise de entidades. Faça as seguintes alterações em:

a. Em Configurações de trabalho, em Nome, insira `reviews-entities-analysis`.

b. Em Configurações de trabalho, em Tipo de análise, escolha Entidades.

c. Em Permissões de acesso, escolha Utilizar um perfil do IAM existente. Em Nome do perfil, escolha `AmazonComprehendServiceRole-comprehend-access-role` (essa é a mesmo perfil que você criou para o trabalho de sentimento).

Analise sentimentos e entidades (AWS CLI)

Você usa os `start-sentiment-detection-job` e os comandos `start-entities-detection-job` para executar trabalhos de análise de sentimentos e entidades. Depois de executar cada comando, ele AWS CLI mostra um objeto JSON com um `JobId` valor que permite acessar detalhes sobre o trabalho, incluindo a localização de saída do S3.

Para executar trabalhos de análise de sentimentos e entidades (AWS CLI)

1. Inicie um trabalho de análise de sentimentos executando o comando a seguir no AWS CLI. Substitua *`arn:aws:iam::123456789012:role/comprehend-access-role`* pelo ARN do perfil do IAM que você copiou anteriormente para um editor de texto. Se sua AWS CLI região padrão for diferente da região na qual você criou seu bucket do Amazon S3, inclua o `--region` parâmetro e *`us-east-1`* substitua pela região em que seu bucket reside.

```
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. Depois de enviar o trabalho, copie o `JobId` e salve-o em um editor de texto. Você precisará do `JobId` para encontrar os arquivos de saída do trabalho de análise.

3. Inicie um trabalho de análise de entidades executando o comando a seguir.

```
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. Depois de enviar o trabalho, copie o `JobId` e salve-o em um editor de texto.
5. Verifique o status dos seus trabalhos. Você pode ver o progresso de um trabalho acompanhando seu `JobId`.

   Para acompanhar o andamento do trabalho de análise de sentimentos, execute o comando a seguir. Substitua *`sentiment-job-id`* pelo `JobId` que você copiou depois de executar sua análise de sentimentos.

   ```
   aws comprehend describe-sentiment-detection-job
   --job-id sentiment-job-id
   ```

   Para monitorar seu trabalho de análise de entidades, execute o comando a seguir. Substitua *`entities-job-id`* pelo `JobId` que você copiou depois de executar sua análise de entidades.

   ```
   aws comprehend describe-entities-detection-job
   --job-id entities-job-id
   ```

   É preciso alguns minutos para que `JobStatus` seja exibido como `COMPLETED`.

Você concluiu trabalhos de análise de sentimentos e entidades. Ambos os trabalhos devem ser concluídos antes de você passar para a próxima etapa. A conclusão do trabalho pode demorar vários minutos.

Etapa 4: preparar a saída do Amazon Comprehend para visualização de dados

Para preparar os resultados das tarefas de análise de sentimentos e entidades para criar visualizações de dados, você usa o AWS Glue e o Amazon Athena. Nesta etapa, você extrai os arquivos de resultados do Amazon Comprehend. Em seguida, você cria um crawler AWS Glue que explora seus dados e os cataloga automaticamente em tabelas no AWS Glue Data Catalog. Depois disso, você acessa e transforma essas tabelas usando Amazon Athena um serviço de consulta interativo e sem servidor. Quando você concluir essa etapa, seus resultados do Amazon Comprehend estarão limpos e prontos para visualização.

Para uma tarefa de detecção de entidades de PII, o arquivo de saída é um texto sem formatação, não um arquivo compactado. O nome do arquivo de saída é o mesmo do arquivo de entrada, com `.out` anexo no final. Não é necessária a etapa de extrair o arquivo de saída. Pule para carregar os dados em um AWS Glue Data Catalog.

Tópicos

- Pré-requisitos
- Baixe a saída
- Extraia os arquivos de saída
- Carregue os arquivos extraídos
- Carregue os dados em um AWS Glue Data Catalog
- Prepare os dados para análise

Pré-requisitos

Antes de começar, conclua Etapa 3: executando trabalhos de análise em documentos no Amazon S3.

Baixe a saída

O Amazon Comprehend usa compressão Gzip para compactar arquivos de saída e salvá-los como um arquivo tar. A maneira mais simples de extrair os arquivos de saída é baixar esses arquivos `output.tar.gz` localmente.

Nesta etapa, você baixará os arquivos de saída de sentimentos e entidades.

Baixe os arquivos de saída (console)

Para encontrar os arquivos de saída para cada tarefa, retorne à tarefa de análise no console do Amazon Comprehend. A tarefa de análise fornece a localização do S3 para a saída, onde você pode baixar o arquivo de saída.

Baixar os arquivos de saída (console)

1. No console do Amazon Comprehend, no painel de navegação, retorne às Tarefas de análise.
2. Escolha sua tarefa de análise de sentimentos `reviews-sentiment-analysis`.

3. Em Saída, escolha o link exibido ao lado do Local dos dados de saída. Isso redireciona você para o arquivo de `output.tar.gz` em seu bucket do S3.
4. Na página Visão geral selecione Fazer download.
5. No seu computador, renomeie o arquivo como `sentiment-output.tar.gz`. Como todos os arquivos de saída têm o mesmo nome, isso ajuda você a acompanhar os arquivos de sentimentos e entidades.
6. Repita as etapas 1 a 4 para encontrar e baixar a saída da sua tarefa de `reviews-entities-analysis`. No seu computador, renomeie o arquivo como `entities-output.tar.gz`.

Baixe os arquivos de saída (AWS CLI)

Para encontrar os arquivos de saída de cada tarefa, use o `JobId` da tarefa de análise para encontrar a localização da saída no S3. Em seguida, use o comando `cp` para baixar o arquivo de saída no seu computador.

Baixar os arquivos de saída (AWS CLI)

1. Para listar detalhes sobre sua tarefa de análise de sentimentos, execute o comando a seguir. Substitua o *`sentiment-job-id`* pelo sentimento `JobId` que você salvou.

   ```
   aws comprehend describe-sentiment-detection-job --job-id sentiment-job-id
   ```

   Se perdeu o controle da sua `JobId`, execute o comando a seguir para listar todas as suas tarefas de sentimento e filtrá-las por nome.

   ```
   aws comprehend list-sentiment-detection-jobs
   --filter JobName="reviews-sentiment-analysis"
   ```

2. No objeto de `OutputDataConfig`, encontre o valor `S3Uri`. O valor `S3Uri` deve ser semelhante ao seguinte formato: *`s3://amzn-s3-demo-bucket/.../output/output.tar.gz`*. Copie esse valor para um editor de texto.
3. Para baixar o arquivo de saída de sentimentos no diretório local, execute o seguinte comando: Substitua o caminho do bucket do S3 pelo `S3Uri` copiado na etapa anterior. Substitua *`path/`* pelo caminho da pasta para seu diretório local. O nome do `sentiment-output.tar.gz` substitui o nome do arquivo original para ajudar você a acompanhar os arquivos de sentimentos e entidades.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz
path/sentiment-output.tar.gz
```

4. Para listar detalhes sobre sua tarefa de análise de entidades, execute o comando a seguir.

```
aws comprehend describe-entities-detection-job
--job-id entities-job-id
```

Se perdeu o controle da sua `JobId`, execute o comando a seguir para listar todas as suas tarefas de entidade e filtrá-las por nome.

```
aws comprehend list-entities-detection-jobs
--filter JobName="reviews-entities-analysis"
```

5. A partir do objeto de `OutputDataConfig` na descrição da tarefa de sua entidade, copie o valor do `S3Uri`.
6. Para baixar o arquivo de saída de entidades no diretório local, execute o seguinte comando. Substitua o caminho do bucket do S3 pelo `S3Uri` copiado na etapa anterior. Substitua *path/* pelo caminho da pasta para seu diretório local. O nome da `entities-output.tar.gz` substitui o nome do arquivo original.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz
path/entities-output.tar.gz
```

Extraia os arquivos de saída

Antes de acessar os resultados do Amazon Comprehend, descompacte os arquivos de sentimentos e entidades. Use o sistema de arquivos local ou um terminal para descompactar os arquivos.

Extraia os arquivos de saída (sistema de arquivos GUI)

Se você usa o macOS, clique duas vezes no arquivo no sistema de arquivos GUI para extrair o arquivo de saída.

Se você usa o Windows, é possível utilizar uma ferramenta de terceiros, como 7-Zip, para extrair os arquivos de saída no sistema de arquivos GUI. No Windows, você deve executar duas etapas para acessar o arquivo de saída. Primeiro, descompacte o arquivo e, em seguida, extraia-o

Renomeie o arquivo de sentimentos como `sentiment-output` e o arquivo de entidades como `entities-output` para distinguir entre os arquivos de saída.

Extraia os arquivos de saída (terminal)

Se você usa Linux ou macOS, é possível utilizar seu terminal padrão. Se você usa o Windows, deve ter acesso a um ambiente no estilo UNIX, como o Cygwin, para executar comandos tar.

Para extrair o arquivo de saída de sentimentos do arquivo de sentimentos, execute o comando a seguir em seu terminal local.

```
tar -xvf sentiment-output.tar.gz --transform 's,^,sentiment-,'
```

Observe que o parâmetro `--transform` adiciona o prefixo `sentiment-` ao arquivo de saída dentro do arquivo, renomeando o arquivo como `sentiment-output`. Isso permite distinguir entre os arquivos de saída de sentimentos e entidades e evitar a substituição.

Para extrair o arquivo de saída de entidades do arquivo de entidades, execute o comando a seguir em seu terminal local.

```
tar -xvf entities-output.tar.gz --transform 's,^,entities-,'
```

O parâmetro `--transform` adiciona o prefixo `entities-` ao nome do arquivo de saída.

> **Tip**
>
> Para economizar custos de armazenamento no Amazon S3, compacte os arquivos novamente com o Gzip antes de carregá-los. É importante descompactar e descompactar os arquivos originais porque não é AWS Glue possível ler automaticamente os dados de um arquivo tar. No entanto, AWS Glue pode ler arquivos no formato Gzip.

Carregue os arquivos extraídos

Depois de extrair os arquivos, faça o upload deles no seu bucket. Você deve armazenar os arquivos de saída de sentimentos e entidades em pastas separadas AWS Glue para ler os dados corretamente. No seu bucket, crie uma pasta para os resultados dos sentimentos extraídos e uma segunda pasta para os resultados das entidades extraídas. É possível criar pastas com o console do Amazon S3 ou com o AWS CLI.

Faça upload de arquivos extraídos para o Amazon S3 (console)

Em seu bucket do S3, crie uma pasta para o arquivo de resultados de sentimento extraído e uma pasta para o arquivo de resultados de entidades. Em seguida, faça o upload dos arquivos de resultados extraídos para suas respectivas pastas.

Fazer o upload dos arquivos extraídos para o Amazon S3 (console)

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Em Buckets, escolha seu bucket e, em seguida, Criar pasta.
3. Para o novo nome da pasta, insira o `sentiment-results` e escolha Salvar. Essa pasta conterá o arquivo de saída do sentimento extraído.
4. Na guia Visão geral do seu bucket, na lista de conteúdos do bucket, escolha a nova pasta `sentiment-results`. Escolha Carregar.
5. Em Adicionar arquivos, escolha o arquivo de `sentiment-output` do seu computador local e selecione Avançar.
6. Deixe as opções de Gerenciar usuários, Acesso para outros Conta da AWS e Gerenciar permissões públicas como padrões. Escolha Próximo.
7. Em Classe de armazenamento, escolha Padrão. Deixe as opções de Criptografia, Metadados e Tag como padrões. Escolha Próximo.
8. Revise as opções de upload e escolha Carregar.
9. Repita as etapas 1 a 8 para criar uma pasta chamada `entities-results` e carregar o arquivo de `entities-output` nela.

Faça o upload dos arquivos extraídos para o Amazon S3 (AWS CLI)

Crie uma pasta no seu bucket do S3 ao fazer o upload de um arquivo com o comando `cp`.

Fazer o upload dos arquivos extraídos no Amazon S3 (AWS CLI)

1. Crie uma pasta de sentimentos e envie seu arquivo de sentimentos para ela executando o comando a seguir. Substitua *`path/`* pelo caminho local para o arquivo de saída de sentimento extraído.

```
aws s3 cp path/sentiment-output s3://amzn-s3-demo-bucket/sentiment-results/
```

2. Crie uma pasta de saída de entidades e carregue seu arquivo de entidades nela executando o comando a seguir. Substitua *path/* pelo caminho local para o arquivo de saída de entidades extraído.

```
aws s3 cp path/entities-output s3://amzn-s3-demo-bucket/entities-results/
```

Carregue os dados em um AWS Glue Data Catalog

Para colocar os resultados em um banco de dados, você pode usar um AWS Glue rastreador. Um AWS Glue rastreador verifica arquivos e descobre o esquema dos dados. Em seguida, ele organiza os dados em tabelas em um AWS Glue Data Catalog (um banco de dados com tecnologia sem servidor). Você pode criar um rastreador com o AWS Glue console ou o. AWS CLI

Carregue os dados em um AWS Glue Data Catalog (console)

Crie um AWS Glue rastreador que `sentiment-results` escaneie suas pastas e `entities-results` pastas separadamente. Um novo perfil de IAM para o AWS Glue concede permissão ao crawler para acessar seu bucket do S3. Esse perfil do IAM é criado ao configurar o crawler.

Para carregar os dados em um AWS Glue Data Catalog (console)

1. Certifique-se de que você esteja em uma região que ofereça suporte AWS Glue. Se estiver em outra região, na barra de navegação, escolha uma região compatível no Seletor de regiões. Para obter uma lista das regiões que oferecem suporte AWS Glue, consulte a Tabela de regiões no Guia Global de Infraestrutura.
2. Abra o AWS Glue console em https://console.aws.amazon.com/glue/.
3. No painel de navegação, escolha Crawlers e Adicionar crawler.
4. Em Nome do crawler, digite `comprehend-analysis-crawler` e escolha Próximo.
5. Em Tipo de fonte do crawler, escolha Armazenamentos de dados e Próximo.
6. Em Adicionar um armazenamento de dados, faça o seguinte:
    a. Em Escolher um armazenamento de dados, escolha S3.
    b. Deixe Conexão em branco.
    c. Na opção Rastrear dados em, escolha Caminho especificado em minha conta.
    d. Em Incluir caminho, insira o caminho do S3 completo da pasta de saída de sentimentos: `s3://amzn-s3-demo-bucket/sentiment-results`.

e. Escolha Próximo.

7. Em Adicionar outro armazenamento de dados, escolha Sim e Próximo. Repita a Etapa 6, mas insira o caminho do S3 completo da pasta de saída das entidades: `s3://amzn-s3-demo-bucket/entities-results`.
8. Em Adicionar outro armazenamento de dados, escolha Não e Próximo.
9. Em Perfil do IAM, siga um destes procedimentos:

   a. Escolha Criar um perfil do IAM.

   b. Para a Perfil do IAM, insira `glue-access-role` e Próximo.
10. Em Criar uma programação para esse crawler, escolha Executar sob demanda e Próximo.
11. Na página Configurar a saída do crawler, faça o seguinte:

    a. Em Base de dados, selecione Adicionar banco de dados.

    b. Em Nome do banco de dados, insira `comprehend-results`. Esse banco de dados armazenará suas tabelas de saída do Amazon Comprehend.

    c. Deixe as outras opções em suas configurações padrão e escolha Próximo.
12. Revise as informações do crawler e escolha Concluir.
13. No console Glue, em Crawlers, escolha `comprehend-analysis-crawler` e Executar crawler. A conclusão do crawler pode levar alguns minutos.

Carregue os dados em um AWS Glue Data Catalog (AWS CLI)

Crie uma função do IAM para AWS Glue que forneça permissão para acessar seu bucket do S3. Em seguida, crie um banco de dados no AWS Glue Data Catalog. Por fim, crie e execute um crawler que carrega seus dados em tabelas no banco de dados.

Para carregar os dados em um AWS Glue Data Catalog (AWS CLI)

1. Para criar uma função do IAM para AWS Glue, faça o seguinte:

   a. Salve a política de confiança a seguir como um documento JSON chamado `glue-trust-policy.json` em seu computador.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Para criar um perfil do IAM, execute o comando a seguir. Substitua *path/* pelo caminho do seu computador local até o documento JSON.

```
aws iam create-role --role-name glue-access-role
--assume-role-policy-document file://path/glue-trust-policy.json
```

c. Quando AWS CLI listar o Amazon Resource Number (ARN) para a nova função, copie e salve-o em um editor de texto.

d. Salve a seguinte política do IAM como um documento JSON chamado `glue-access-policy.json` em seu computador. A política concede AWS Glue permissão para rastrear suas pastas de resultados.

```
{
    "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. Para criar uma política do IAM, execute o comando a seguir. Substitua *path/* pelo caminho do seu computador local até o documento JSON.

```
aws iam create-policy --policy-name glue-access-policy
```

```
--policy-document file://path/glue-access-policy.json
```

f. Quando AWS CLI listar o ARN da política de acesso, copie e salve em um editor de texto.

g. Anexe a nova política ao perfil do IAM executando o comando a seguir. Substitua *policy-arn* pelo ARN da política do IAM anotado na etapa anterior.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name glue-access-role
```

h. Anexe a política AWS gerenciada `AWSGlueServiceRole` à sua função do IAM executando o comando a seguir.

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
--role-name glue-access-role
```

2. Crie um AWS Glue banco de dados executando o comando a seguir.

```
aws glue create-database
--database-input Name="comprehend-results"
```

3. Crie um novo AWS Glue rastreador executando o comando a seguir. *glue-iam-role-arn*Substitua pelo ARN da sua AWS Glue função do 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. Inicie o crawler executando o comando a seguir:

```
aws glue start-crawler --name comprehend-analysis-crawler
```

A conclusão do crawler pode levar alguns minutos.

Prepare os dados para análise

Agora você tem um banco de dados preenchido com os resultados do Amazon Comprehend. No entanto, os resultados estão aninhados. Para desaninhá-los, você executa algumas instruções SQL em Amazon Athena. Amazon Athena é um serviço de consulta interativo que facilita a análise de dados no Amazon S3 usando SQL padrão. O Athena não tem servidor, portanto, não há infraestrutura para gerenciar e tem um modelo de preços. pay-per-query Nesta etapa, você cria novas tabelas de dados limpos que podem ser usadas para análise e visualização. Você usa o console do Athena para prepará-los.

Preparar os dados

1. Abra o console do Athena em https://console.aws.amazon.com/athena/.
2. No editor de consultas, escolha Configurações e escolha Gerenciar.
3. Em Localização dos resultados da consulta, insira `s3://amzn-s3-demo-bucket/query-results/`. Isso cria uma nova pasta chamada `query-results` em seu bucket que armazena a saída das Amazon Athena consultas que você executa. Escolha Salvar.
4. No editor de consultas, escolha Editor.
5. Em Banco de dados, escolha o AWS Glue banco de dados `comprehend-results` que você criou.
6. Na seção Tabelas, haverá duas tabelas chamadas `sentiment_results` e `entities_results`. Visualize as tabelas para garantir que o crawler tenha carregado os dados. Nas opções de cada tabela (os três pontos ao lado do nome da tabela), escolha Visualizar tabela. Uma consulta curta é executada automaticamente. Verifique o painel Resultados para garantir que as tabelas contenham dados.

   > **Tip**
   >
   > Se as tabelas não tiverem dados, tente verificar as pastas em seu bucket do S3. Certifique-se de haver uma pasta para resultados de entidades e uma pasta para resultados de sentimentos. Em seguida, tente executar um novo AWS Glue rastreador.

7. Para desaninhar a tabela de `sentiment_results`, insira a seguinte consulta no Editor de consultas e escolha Executar.

```
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. Para começar a desaninhar a tabela de entidades, insira a consulta a seguir no Editor de consultas e escolha Executar.

```
CREATE TABLE entities_results_1 AS
SELECT file, line, nested FROM entities_results
CROSS JOIN UNNEST(entities) as t(nested)
```

9. Para concluir o desaninhamento da tabela de entidades, insira a consulta a seguir no Editor de consultas e escolha Executar consulta.

```
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
```

Sua tabela de `sentiment_results_final` deve ter a aparência a seguir, com colunas denominadas arquivo, linha, sentimento, mista, negativa, neutra e positiva. A tabela deve ter um valor por célula. A coluna de sentimentos descreve o sentimento geral mais provável de uma avaliação específica. As colunas mista, negativa, neutra e positiva dão pontuações para cada tipo de sentimento.

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	6.536858109029708 4E-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

Sua tabela de `entities_results_final` deve ter a aparência a seguir, com colunas chamadas arquivo, linha, beginoffset, endoffset, pontuação, entidade e categoria. A tabela deve ter um valor por célula. A coluna de pontuação indica a confiança do Amazon Comprehend na entidade detectada. A categoria indica que tipo de entidade o Comprehend detectou.

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

Agora que tem os resultados do Amazon Comprehend carregados em tabelas, você pode visualizar e extrair insights significativos dos dados.

Etapa 5: visualizando o resultado do Amazon Comprehend em QuickSight

Depois de armazenar os resultados do Amazon Comprehend em tabelas, você pode se conectar e visualizar os dados com. QuickSight QuickSight é uma ferramenta AWS gerenciada da de business intelligence (BI) para visualização de dados. QuickSightfacilita a conexão com sua fonte de dados e a criação de recursos visuais potentes. Nesta etapa, você se conecta QuickSight aos seus dados, cria visualizações que extraem insights dos dados e publica um painel de visualizações.

Tópicos

- Pré-requisitos
- Concessão de QuickSight acesso
- Importe os conjuntos de dados
- Crie uma visualização de sentimentos
- Crie uma visualização de entidades
- Publique um painel
- Limpeza

Pré-requisitos

Antes de começar, conclua Etapa 4: preparar a saída do Amazon Comprehend para visualização de dados.

Concessão de QuickSight acesso

Para importar os dados, é QuickSight necessário acesso ao bucket e às tabelas do Amazon Simple Storage Service (Amazon S3). Amazon Athena Para dar QuickSight acesso aos seus dados, você deve estar conectado como QuickSight administrador e ter acesso para editar as permissões do recurso. Se você não conseguir concluir as etapas a seguir, revise os pré-requisitos do IAM na página de visão geral Tutorial: analisando insights de avaliações de clientes com o Amazon Comprehend.

Para dar QuickSight acesso aos seus dados

1. Abra o console de QuickSight.
2. Se for a primeira vez que você usa QuickSight, o console solicita que você crie um novo usuário administrador fornecendo um endereço de e-mail. Para Endereço de E-mail, insira o mesmo endereço de e-mail da sua Conta da AWS. Escolha Continuar.

3. Depois de entrar, escolha o nome do seu perfil na barra de navegação e escolha Gerenciar QuickSight. Você deve estar conectado como administrador para ver a QuickSight opção Gerenciar.
4. Escolha Segurança e permissões.
5. Para QuickSight acessar os AWS serviços, escolha Adicionar ou remover.
6. Escolha Amazon S3.
7. Em Selecionar buckets do Amazon S3, escolha seu bucket do S3 em S3 Bucket e Permissão de gravação para o Athena Workgroup.
8. Escolha Terminar.
9. Selecione Atualizar.

Importe os conjuntos de dados

Antes de criar visualizações, você deve adicionar os conjuntos de dados de sentimentos e entidades ao. QuickSight Você faz isso com o QuickSight console. Você importa suas tabelas de sentimentos e entidades desaninhadas do. Amazon Athena

Para importar seus conjuntos de dados

1. Abra o console de QuickSight.
2. Na barra de navegação, em Conjuntos de dados, escolha Novo conjunto de dados.
3. Em Criar um conjunto de dados, escolha Athena.
4. EmNome da fonte de dados, insira `reviews-sentiment-analysis` e escolha Criar fonte de dados.
5. Em Banco de dados, selecione o banco de dados `comprehend-results`.
6. Em Tabelas, escolha a tabela de sentimentos `sentiment_results_final` e, em seguida, escolha Selecionar.
7. Escolha Importar para SPICE para acelerar a análise e escolha Visualizar. O SPICE QuickSight é o mecanismo de cálculo em memória que fornece análises mais rápidas do que a consulta direta ao criar visualizações.
8. Volte ao QuickSight console e escolha Conjuntos de dados. Repita as etapas de 1 a 7 para criar um conjunto de dados de entidades, mas faça as seguintes alterações:

    a. Para Nome da fonte de dados, insira `reviews-entities-analysis`.

b. Em Tabelas, escolha a tabela de entidades `entities_results_final`.

Crie uma visualização de sentimentos

Agora que você pode acessar seus dados QuickSight, você pode começar a criar visualizações. Você cria um gráfico circular com os dados de sentimento do Amazon Comprehend. O gráfico circular mostra a proporção das avaliações positivas, neutras, mistas e negativas.

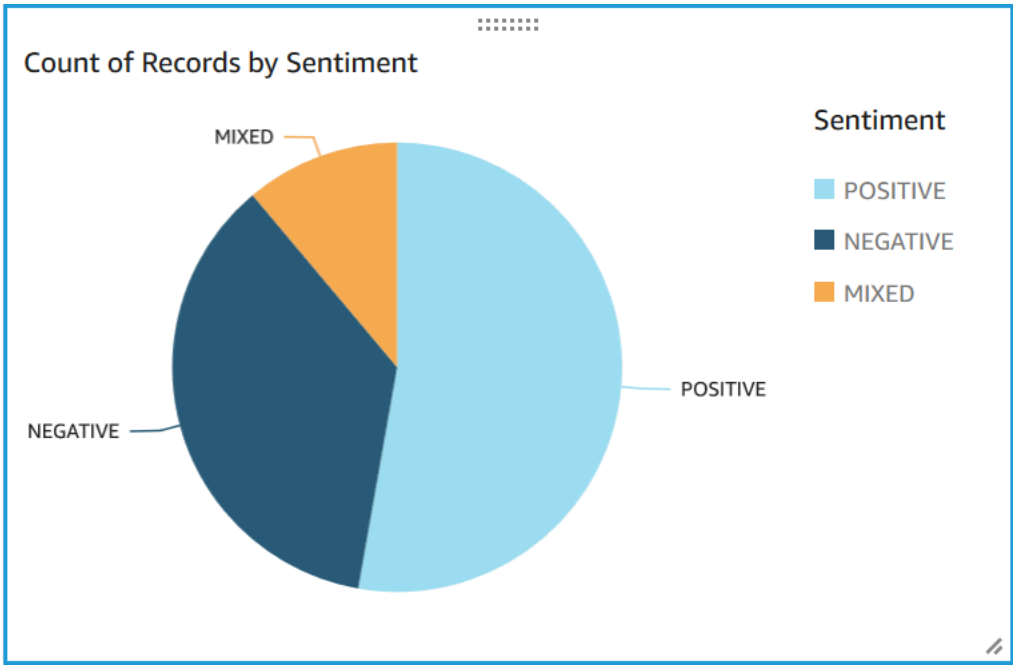
Para visualizar dados de sentimentos

1. No QuickSight console, escolha Análises e, em seguida, escolha Nova análise.
2. Em Seus conjuntos de dados, escolha o conjunto de dados de sentimentos `sentiment_results_final` e, em seguida, escolha Criar análise.
3. No editor visual, na Lista de campos, escolha sentimento.

> **Note**
>
> Os valores na Lista de campos dependem dos nomes das colunas usadas para criar as tabelas em Amazon Athena. Se você alterou os nomes das colunas fornecidos nas consultas SQL, os nomes da Lista de campos serão diferentes dos nomes usados nesses exemplos de visualização.

4. Para Tipos de visuais, escolha Gráfico de pizza.

Um gráfico circular semelhante ao seguinte, com seções positivas, neutras, mistas e negativas, é exibido. Para ver a contagem e a porcentagem de uma seção, passe o mouse sobre ela.



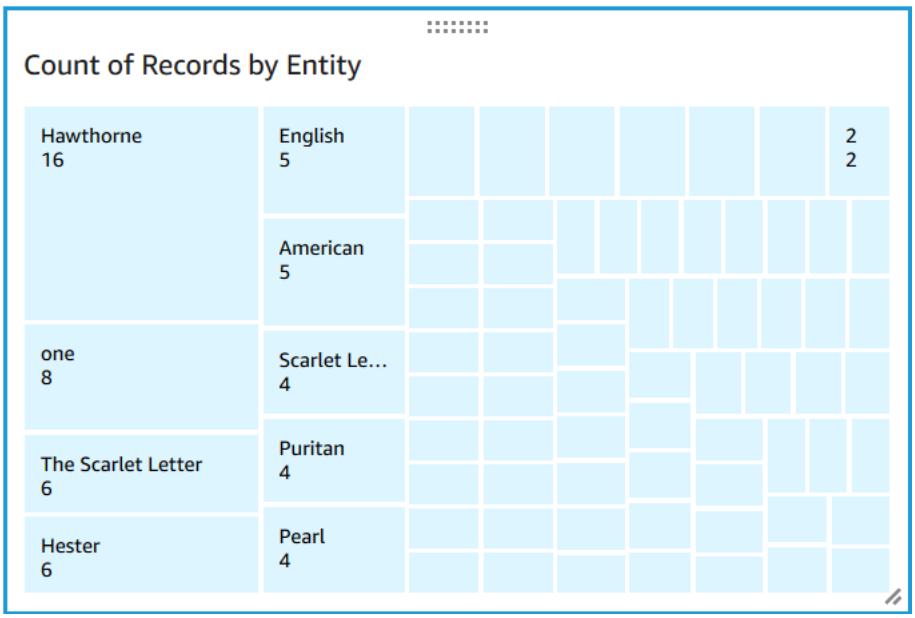
Crie uma visualização de entidades

Agora, crie uma segunda visualização com o conjunto de dados de entidades. Você cria um mapa de árvore das entidades distintas nos dados. Cada bloco no mapa de árvore representa uma entidade e o tamanho do bloco se correlaciona com o número de vezes que a entidade aparece no conjunto de dados.

Para visualizar dados de entidades

1. No painel de controle Visualizar, ao lado de Conjunto de dados, escolha o ícone Adicionar, editar, substituir e remover conjuntos de dados.
2. Escolha Adicionar conjunto de dados.
3. Em Escolher conjunto de dados a ser adicionado, escolha o conjunto de dados `entities_results_final` de suas entidades na lista de conjuntos de dados e escolha Selecionar.
4. No painel de controle Visualizar, escolha a lista suspensa Conjunto de dados e escolha o conjunto de dados de entidades `entities_results_final`.
5. Na Lista de campos, escolha entidade.
6. Para Tipos de visuais, escolha Mapa de árvore.

Um mapa de árvore semelhante ao seguinte é exibido ao lado do seu gráfico circular. Para ver a contagem de uma entidade específica, passe o mouse sobre um bloco.



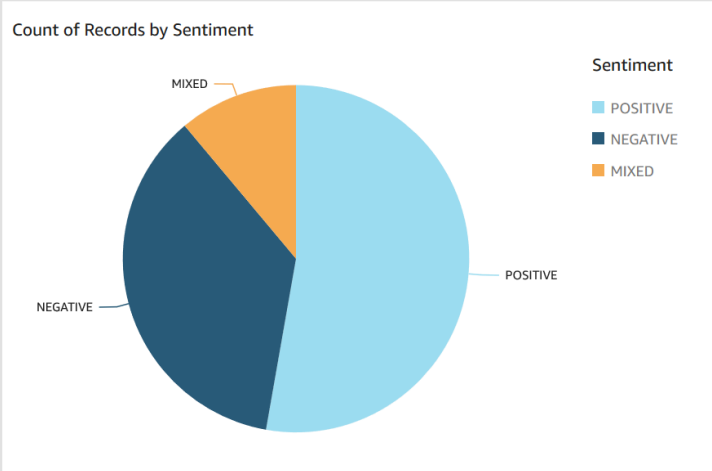
Publique um painel

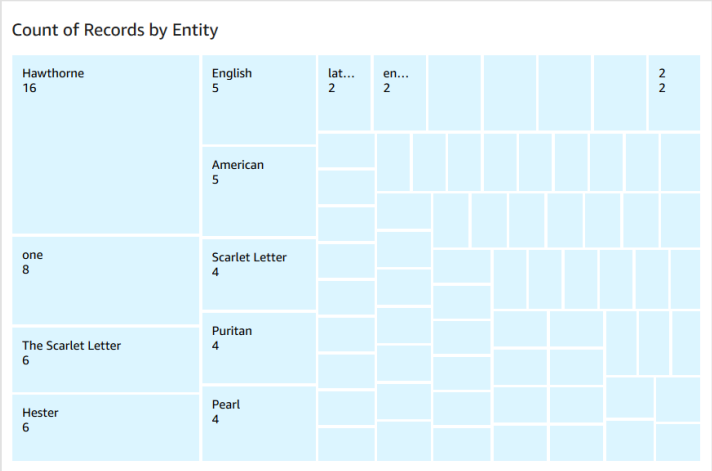
Depois de criar as visualizações, você pode publicá-las como um painel. Você pode realizar várias tarefas com um painel, como compartilhá-lo com os usuários da sua Conta da AWS, salvá-lo como PDF ou enviá-lo por e-mail como um relatório (limitado à Enterprise Edition do QuickSight). Nesta etapa, você publica as visualizações como um painel na sua conta.

Para publicar seu painel

1. Na barra de navegação, escolha Compartilhar.
2. Escolha Publicar painel.
3. Escolha Publicar novo painel como e insira o nome `comprehend-analysis-reviews` para o painel.
4. Escolha Publicar painel.
5. Feche o painel Compartilhar painel com os usuários escolhendo o botão de fechar no canto superior direito.
6. No QuickSight console, no painel de navegação, escolha Painéis. Uma miniatura do seu novo painel `comprehend-analysis-reviews` deve aparecer em Painéis. Escolha o painel para visualizá-lo.

Agora você tem um painel com visualizações de sentimentos e entidades que se parece com o exemplo a seguir.





> **Tip**
>
> Se você quiser editar as visualizações em seu painel, volte para Análises e edite a visualização que você deseja atualizar. Em seguida, publique o painel novamente como um novo painel ou como um substituto do painel existente.

Limpeza

Depois de concluir este tutorial, talvez você queira limpar os AWS recursos da que não queira mais usar. Os AWS recursos da ativos podem continuar gerando cobranças em sua conta.

As ações a seguir podem ajudar a evitar cobranças contínuas:

- Cancele sua QuickSight assinatura. QuickSight é um serviço de assinatura mensal. Para cancelar sua assinatura, consulte Cancelar sua assinatura no Guia do QuickSight usuário.
- Exclua seu bucket do Amazon S3. O Amazon S3 cobra pelo armazenamento. Para limpar seus recursos do Amazon S3, exclua seu bucket. Para obter mais informações sobre como excluir um bucket, consulte Como excluir um bucket do S3? no Guia do usuário do Amazon Simple Storage Service. Certifique-se de que você salvou todos os seus arquivos importantes antes de excluir seu bucket.
- Limpe seu AWS Glue Data Catalog. O é AWS Glue Data Catalog cobrado mensalmente. Você pode excluir seus bancos de dados para evitar cobranças contínuas. Para obter informações sobre como gerenciar seus AWS Glue Data Catalog bancos de dados, consulte Como trabalhar com

bancos de dados no AWS Glue console no Guia do AWS Glue desenvolvedor. Certifique-se de que você exportou seus dados antes de limpar qualquer banco de dados ou tabela.

Utilizar pontos de acesso do Amazon S3 object Lambda para informações de identificação pessoal (PII)

Você pode usar os pontos de acesso do Amazon S3 Object Lambda para informações de identificação pessoal (PII) a fim de configurar como os documentos são recuperados do seu bucket do Amazon S3. Você pode controlar o acesso a documentos que contêm PII e editar PIIs nos documentos. Para obter mais informações sobre como o Amazon Comprehend pode detectar PIIs em seus documentos, consulte Detectar entidades de PII. Os pontos de acesso Amazon S3 Object Lambda usam AWS Lambda funções para transformar automaticamente a saída de uma solicitação GET padrão do Amazon S3. Para obter mais informações, consulte Transformando objetos com o objeto Lambda do Amazon S3 no Guia do usuário do Amazon Simple Storage Service.

Quando você cria um ponto de acesso do Amazon S3 Object Lambda para PII, os documentos são processados usando as funções do Lambda do Amazon Comprehend para controlar o acesso a documentos que contêm PII e editar PIIs em documentos.

Quando você cria um ponto de acesso do Amazon S3 Object Lambda para PII, os documentos são processados usando as seguintes funções do Lambda do Amazon Comprehend:

- `ComprehendPiiAccessControlS3ObjectLambda`: controla o acesso a documentos com PII armazenadas no bucket do S3. Para obter mais informações sobre essa função Lambda, faça login no AWS Management Console para ver a ObjectLambda função ComprehendPiiAccessControlS3 no. AWS Serverless Application Repository
- `ComprehendPiiRedactionS3ObjectLambda`: edita as PIIs de documentos no bucket do Amazon S3. Para obter mais informações sobre essa função Lambda, faça login no AWS Management Console para ver a ObjectLambda função ComprehendPiiRedactionS3 no. AWS Serverless Application Repository

Para obter informações sobre como implantar aplicativos sem servidor a partir do AWS Serverless Application Repository, consulte Implantação de aplicativos no Serverless AWS Application Repository Developer Guide.

Tópicos

- Controlar o acesso a documentos com informações de identificação pessoal (PII)
- Editar informações de identificação pessoal (PII) de documentos

Controlar o acesso a documentos com informações de identificação pessoal (PII)

É possível usar um ponto de acesso do Amazon S3 Object Lambda para controlar o acesso a documentos com informações de identificação pessoal (PII).

Para garantir que só os usuários autorizados tenham acesso aos documentos que contêm PII armazenadas em seu bucket do Amazon S3, você usa a função `ComprehendPiiAccessControlS3ObjectLambda`. Essa função Lambda usa a ContainsPiiEntitiesoperação ao processar uma solicitação GET padrão do Amazon S3 em objetos de documento.

Por exemplo, se você tiver documentos no bucket do S3 que incluam PII, como números de cartões de crédito ou informações de contas bancárias, você poderá configurar a função `ComprehendPiiAccessControlS3ObjectLambda` para detectar esses tipos de entidade de PII e restringir o acesso a usuários não autorizados. Para obter mais informações sobre os tipos de entidades de PII compatíveis, consulte Tipos de entidades universais de PII.

Para obter mais informações sobre essa função Lambda, faça login no AWS Management Console para ver a ObjectLambda função ComprehendPiiAccessControlS3 no. AWS Serverless Application Repository

Criar um ponto de acesso do Amazon S3 object Lambda para controlar o acesso a documentos

O exemplo abaixo cria um ponto de acesso do Amazon S3 Object Lambda para controlar o acesso a documentos que contêm números de previdência social.

Criação de um ponto de acesso Lambda do objeto Amazon S3 usando o AWS Command Line Interface

Crie uma configuração para o ponto de acesso do Amazon S3 Object Lambda e salve a configuração em um arquivo chamado 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\"}"
                }
            }
        }
    ]
}
```

O exemplo a seguir cria um ponto de acesso do Amazon S3 Object Lambda com base na configuração definida no arquivo `config.json`.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws s3control create-banner-access-point \
    --region region \
    --account-id account-id \
    --name s3-object-lambda-access-point \
    --configuration file://config.json
```

Invocar um ponto de acesso do Amazon S3 object Lambda para controlar o acesso a documentos

O exemplo abaixo invoca um ponto de acesso do Amazon S3 Object Lambda para controlar o acesso a documentos.

Invocando um ponto de acesso Lambda do objeto Amazon S3 usando o AWS Command Line Interface

O exemplo a seguir invoca um ponto de acesso do Amazon S3 Object Lambda usando o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws s3api get-object \
    --region region \
    --bucket s3-object-lambda-access-point-name-arn \
    --key object-prefix-key output-file-name
```

Editar informações de identificação pessoal (PII) de documentos

Você pode usar um ponto de acesso do Amazon S3 Object Lambda para redigir informações de identificação pessoal (PII) de documentos.

Para editar os tipos de entidades de PII de documentos armazenados em um bucket do S3, você usa a função `ComprehendPiiRedactionS3ObjectLambda`. Essa função Lambda usa as DetectPiiEntitiesoperações ContainsPiiEntitiese ao processar uma solicitação GET padrão do Amazon S3 em objetos de documento.

Por exemplo, se os documentos em seu bucket do S3 incluem PIIs, como números de cartão de crédito ou informações de contas bancárias, você poderá configurar a função `ComprehendPiiRedactionS3ObjectLambda` para detectar PIIs e, em seguida, retornar uma cópia desses documentos nas quais os tipos de entidade de PII estejam editados. Para obter mais informações sobre os tipos de entidades de PII compatíveis, consulte Tipos de entidades universais de PII.

Para obter mais informações sobre essa função Lambda, faça login no AWS Management Console para ver a ObjectLambda função ComprehendPiiRedactionS3 no. AWS Serverless Application Repository

Criar um ponto de acesso do Amazon S3 object Lambda para redigir PIIs em documentos

O exemplo abaixo cria um ponto de acesso do Amazon S3 Object Lambda para redigir números de cartões de crédito de documentos.

Criação de um ponto de acesso Lambda do objeto Amazon S3 usando o AWS Command Line Interface

Crie uma configuração de ponto de acesso do Amazon S3 Object Lambda e salve a configuração em um arquivo chamado `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
\"}"
                }
            }
        }
    ]
}
```

O exemplo abaixo demonstra a criação de um ponto de acesso do Amazon S3 Object Lambda com base na configuração definida no `config.json`

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws s3control create-access-point-for-object-lambda \
    --region region \
    --account-id account-id \
    --name s3-object-lambda-access-point \
    --configuration file://config.json
```

Invocar um ponto de acesso do Amazon S3 object Lambda para redigir PIIs em documentos

Os exemplos abaixo invocam um ponto de acesso do Amazon S3 Object Lambda para redigir PIIs em documentos.

Invocando um ponto de acesso Lambda do objeto Amazon S3 usando o AWS Command Line Interface

O exemplo a seguir invoca um ponto de acesso do Amazon S3 Object Lambda usando o AWS CLI.

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws s3api get-object \
    --region region \
    --bucket s3-object-lambda-access-point-name-arn \
    --key object-prefix-key output-file-name
```

Solução: analisar texto com o Amazon Comprehend e OpenSearch

AWS fornece uma implementação de referência da análise de texto usando o Amazon Comprehend e o serviço. OpenSearch O Amazon Comprehend fornece análise de texto e fornece indexação, pesquisa OpenSearch e visualização de documentos.

Para obter mais informações, consulte Análise de texto com o OpenSearch Amazon Comprehend.

Referência da API

A referência de API agora é um documento separado. Para mais informações, consulte Referência de API do Amazon Comprehend.

Histórico do documento do Amazon Comprehend

A tabela a seguir descreve a documentação para esta versão do Amazon Comprehend.

Alteração	Descrição	Data
Treinamento personalizado de classificadores com documentos nativos	O Amazon Comprehend agora oferece suporte ao treinamento personalizado de classificadores com documentos nativos. Para obter mais informações, consulte Modelos de classificação de treinamento no Amazon Comprehend.	19 de abril de 2023
Flywheels para gerenciar modelos personalizados	Agora, o Amazon Comprehend oferece suporte a flywheels para ajudar você a gerenciar o treinamento e o monitorar as versões de modelos para modelos personalizados. Para obter mais informações, consulte Flywheels no Amazon Comprehend.	28 de fevereiro de 2023
Tópicos de segurança do IAM atualizados	Os tópicos de segurança do IAM foram atualizados para incluir identidades federadas. Para obter mais informações, consulte Identity and Access Management para o Amazon Comprehend.	22 de dezembro de 2022
Processamento em uma etapa para inferência com modelos personalizados	Agora, o Amazon Comprehend executa automaticamente a extração de texto para	1º de dezembro de 2022

	documentos de entrada de imagem, PDF ou Word antes de executar a classific ação personalizada ou o reconhecimento personali zado de entidades. Para obter mais informações, consulte Processamento de documento s no Amazon Comprehend.	
Síncrono APIs para um sentimento direcionado	O Amazon Comprehend agora oferece suporte à análise APIs síncrona e em tempo real do console para o sentiment o direcionado. Sentiment os direcionados determina m o sentimento associado a entidades específicas em um documento. Para obter mais informações, consulte Sentimentos direcionados no Amazon Comprehend.	21 de setembro de 2022

Anotações mínimas mais baixas para reconhecedores de treinamento	O Amazon Comprehend reduziu os requisitos mínimos para treinar um reconhecedor com arquivos de anotação CSV em texto simples. Agora você pode criar um modelo personalizado de reconhecimento de entidades com apenas três documentos anotados e, pelo menos, 25 anotações por tipo de entidade. Para obter mais informações, consulte Preparando os dados de treinamento.	3 de agosto de 2022
Maior tamanho do documento de entrada em tempo real APIs	O Amazon Comprehend agora oferece suporte a um documento de entrada de até 100 KB na maior parte do tempo real. APIs Para obter mais informações, consulte Diretrizes e cotas.	18 de julho de 2022
Tipos adicionais de entidades de PII	Tipos adicionais de entidades de PII agora detectados pelo Amazon Comprehend. Para obter mais informações, consulte Detectando entidades PII no Amazon Comprehend.	20 de maio de 2022

Reestruturação do índice	Reestruturação do índice do Guia do desenvolvedor do Amazon Comprehend para facilitar a navegação. Para obter mais informações, consulte O que é a Amazon Comprehend.	7 de abril de 2022
Sentimento direcionado	Agora, o Amazon Comprehend oferece suporte à análise de sentimentos direcionados, que determina o sentimento associado a entidades específicas em um documento. Para obter mais informações, consulte Sentimentos direcionados no Amazon Comprehend.	9 de março de 2022
Novo atributo	Agora, o Amazon Comprehend permite que você analise imagens para reconhecimento personalizado de entidades. Para obter mais informações, consulte Detecção de entidades personalizadas no Amazon Comprehend.	28 de fevereiro de 2022
Novo recurso	Agora, você pode copiar modelos personalizados treinados entre Contas da AWS. Para obter mais informações, consulte Copiando modelos personalizados entre contas no Amazon Comprehend.	2 de fevereiro de 2022

Novo recurso	Agora você pode usar AWS Trusted Advisor para visualizar recomendações que podem ajudá-lo a otimizar o custo e a segurança dos seus endpoints do Amazon Comprehend. Para obter mais informações, consulte Utilizar Trusted Advisor com o Amazon Comprehend.	29 de setembro de 2021
Novo recurso	O Amazon Comprehend lançou um conjunto de atributos para o Comprehend Custom que permite melhorias contínuas de modelos, oferecendo a capacidade de criar novas versões do modelo, testar continuamente em conjuntos de testes específicos e realizar a migração ao vivo para novos endpoints do modelo.	21 de setembro de 2021
Novo recurso	Agora, o Amazon Comprehend permite que você analise documentos em PDF e Word para reconhecimento personalizado de entidades. Com os formatos PDF e Word, você pode extrair informações de documentos contendo cabeçalhos, listas e tabelas.	14 de setembro de 2021

Novo recurso	O Amazon Comprehend lançou um novo atributo de visão geral de endpoints que fornece uma visão global dos seus endpoints. Na página de visão geral dos endpoints , você pode visualizar todos os seus endpoints em um só lugar para entender o uso do endpoint em comparação com o uso real dos recursos.	24 de agosto de 2021
Novo recurso	Agora, o Amazon Comprehend Medical permite que você estabeleça uma conexão privada com a sua nuvem privada virtual (VPC) criando um endpoint da VPC de interface. Para obter mais informações, consulte VPC endpoints (). PrivateLink	13 de junho de 2021
Expansão do idioma	O Amazon Comprehend adicionou outros quatro idiomas para o atributo de idioma dominante: hausa (ha), lao (lo), maltês (mt) e oromo (om). Para obter mais informações, consulte Idiomas compatíveis no Amazon Comprehend.	10 de maio de 2021

Novo recurso	Agora, com o Amazon Comprehend, você pode criptografar modelos personalizados usando uma chave gerenciada pelo cliente (CMK). Para obter mais informações, consulte Criptografia do KMS no Amazon Comprehend.	31 de março de 2021
Novo recurso	Agora, você pode usar os pontos de acesso do Amazon S3 Object Lambda para configurar como os documentos que contêm informações de identificação pessoal (PII) são recuperados do seu bucket do Amazon S3. Você pode controlar o acesso a documentos que contêm PII e editar PIIs de documentos. Para obter mais informações, consulte Utilizar pontos de acesso do Amazon S3 Object Lambda para informações de identificação pessoal (PII).	18 de março de 2021

Novo recurso	Agora, é possível rotular um documento com informaçõ es de identificação pessoal (PII). O Amazon Comprehend pode analisar seu documento para detectar a presença de PIIs e devolver os rótulos dos tipos de entidade de PII identificados, como nome, endereço, número da conta bancária ou número de telefone. Para obter mais informações, consulte Rotule um documento com PIIs.	11 de março de 2021
Novo recurso	Agora, com o Amazon Comprehend, você pode detectar eventos em um conjunto de documentos. Quando você cria um trabalho de detecção de eventos assíncronos, o Amazon Comprehend pode detectar tipos compatíveis de eventos financeiros. Para obter mais informações, consulte Detecte eventos.	24 de novembro de 2020

Novo recurso	Agora, o Amazon Comprehend permite que você use o ajuste de escala automático para endpoints personalizados de reconhecimento de entidades. Com o ajuste de escala automático, você pode definir automaticamente o provisionamento de endpoints para atender às suas necessidades de capacidade. Para obter mais informações, consulte Ajuste de escala automática com endpoints.	28 de setembro de 2020
Novo recurso	Para treinar classificadores personalizados ou reconhecedores de entidades, agora você pode fornecer arquivos de manifesto aumentados, que são conjuntos de dados rotulados produzidos pelo Amazon AI Ground Truth. SageMaker Para obter mais informações sobre esses arquivos e, por exemplo, ver o Modo multiclasse, Modo multirrótulo e Anotações.	22 de setembro de 2020

Novo tutorial	Agora, o Amazon Comprehend tem um tutorial que orienta você através de um fluxo de trabalho de vários serviços, analisando avaliações de clientes e visualizando os resultados da análise. Para obter mais informações, consulte Tutorial: Analisando insights de avaliações.	17 de setembro de 2020
Novo recurso	Agora, com o Amazon Comprehend, você pode detectar entidades em seu texto que contêm informações de identificação pessoal (PII), como endereços, números de contas bancárias ou números de telefone. O Amazon Comprehend pode fornecer a localização de cada entidade de PII em seu texto ou pode fornecer uma cópia do seu texto no qual a PII é editada. Para obter mais informações, consulte Detecte informações de identificação pessoal (PII).	17 de setembro de 2020

Novo recurso	Anteriormente, você só podia treinar um modelo em até 12 entidades personalizadas. Agora, o Amazon Comprehend permite que você treine um modelo em até 25 entidades personalizadas por vez. Para obter mais informações, consulte Reconhecimento personalizado de entidades.	12 de agosto de 2020
Expansão do idioma	O Amazon Comprehend adicionou outros cinco idiomas para o atributo personalizado de reconhecimento de entidades: alemão (de), espanhol (es), francês (fr), italiano (it) e português (pt). Para obter mais informações, consulte Idiomas compatíveis no Amazon Comprehend.	12 de agosto de 2020
Novo recurso	Agora, o Amazon Comprehend permite que você estabeleça uma conexão privada entre a sua nuvem privada virtual (VPC) criando um endpoint da VPC de interface. Para obter mais informações, consulte VPC endpoints ().AWS PrivateLink	11 de agosto de 2020

Novo recurso	Agora, com o Amazon Comprehend, você pode detectar rapidamente entidades personalizadas em documentos de texto individuais executando análises em tempo real. Para obter mais informações, consulte Detectando entidades personalizadas em tempo real com o Amazon Comprehend.	9 de julho de 2020
Novo atributo adicionado	Agora, o Amazon Comprehend fornece suporte para um segundo modo de classificação personalizada assíncrona para documentos, o que proporciona maior flexibilidade ao aplicar classes personalizadas a documentos. Enquanto o modo multiclasse associa apenas uma única classe a cada documento, o novo modo de vários rótulos pode associar mais de uma. Por exemplo, um filme pode ser classificado como ficção científica e ação ao mesmo tempo. Para obter mais informações, consulte Modos multiclasse e multirrótulos na classificação personalizada.	19 de dezembro de 2019

Novo atributo adicionado	Agora, o Amazon Comprehend fornece suporte para a classificação personalizada em tempo real de documentos com texto não estruturado. Os clientes podem usar a classificação personalizada em tempo real para entender, rotular e rotear informações com base em suas próprias regras comerciais, de forma síncrona. Para obter mais informações, consulte Análise em tempo real com classificação personalizada.	25 de novembro de 2019
Novo idiomas adicionados	O Amazon Comprehend adicionou outros seis idiomas para vários de seus atributos: árabe (ar), hindi (hi), japonês (ja), coreano (ko), chinês simplificado (zh) e chinês tradicional (zh-TW). Esses novos idiomas só são compatíveis com as operações de determinar sentimento, detectar frases-chave e detecção não personalizada de entidades. Para obter mais informações, consulte Idiomas compatíveis.	6 de novembro de 2019

Novo recurso	Anteriormente, você só podia treinar um modelo em uma única entidade personalizada. Como resultado, você só poderia pesquisar aquela entidade com uma operação de reconhecimento de entidade. O Amazon Comprehend mudou isso e, agora, você pode treinar um modelo em até 12 entidades personalizadas por vez. Para obter mais informações, consulte Reconhecimento personalizado de entidades	9 de julho de 2019
Novo recurso	Agora, o Amazon Comprehend fornece uma matriz de confusão multiclasse para aumentar a capacidade de analisar métricas ao treinar um classificador personalizado. Atualmente, isso é suportado usando o APIs único. Para obter mais informações, consulte Marcação de recursos no Amazon Comprehend.	5 de abril de 2019

Novo recurso	O Amazon Comprehend fornece tags para classificadores personalizados e reconhecedores personalizados de entidades, que podem ser usadas como metadados que permitem que você organize, filtre e controle o acesso aos seus recursos com um nível de controle mais preciso do que nunca. Para obter mais informações, consulte Marcação de recursos no Amazon Comprehend.	3 de abril de 2019
Novo recurso	O Amazon S3 já permite que você criptografe seus documentos de entrada, e o Amazon Comprehend estende isso ainda mais. Ao usar sua própria chave do KMS, você pode não apenas criptografar os resultados de saída do seu trabalho, mas, também, os dados sobre o volume de armazenamento anexado à instância de computação que processa o trabalho de análise. O resultado é end-to-end segurança. Para obter mais informações, consulte Criptografia do KMS no Amazon Comprehend	28 de março de 2019

Novo recurso	O reconhecimento personalizado de entidades amplia a capacidade do Amazon Comprehend, permitindo que você identifique novos tipos de entidade não compatíveis, como um dos tipos de entidade genéricos predefinidos. Isso significa que você pode analisar documentos e extrair entidades, como códigos de produtos ou entidades específicas a negócios, que atendam às suas necessidades específicas. Para obter mais informações, consulte Reconhecimento personalizado de entidades	16 de novembro de 2018
Novo recurso	Você pode usar o Amazon Comprehend para compilar seus próprios modelos para classificação personalizada, atribuindo um documento a uma classe ou a uma categoria. Para obter mais informações, consulte Classificação de documentos.	15 de novembro de 2018
Expansão de região	Agora, o Amazon Comprehend está disponível na Europa (Frankfurt) (eu-central-1).	10 de outubro de 2018

Expansão do idioma	Além do inglês e do espanhol, agora, o Amazon Comprehend também pode examinar documentos em francês, alemão, italiano e português. Para obter mais informações, consulte Idiomas compatíveis no Amazon Comprehend.	10 de outubro de 2018
Expansão de região	Agora, o Amazon Comprehend está disponível na Ásia-Pacífico (Sydney) (ap-southeast-2).	15 de agosto de 2018
Novo recurso	Agora, o Amazon Comprehend analisa documentos para descobrir a sintaxe de um documento e a parte da fala de cada palavra. Para obter mais informações, consulte Sintaxe.	17 de julho de 2018
Novo recurso	Agora, o Amazon Comprehend oferece suporte ao processamento assíncrono em lotes para detecção de idioma, frase-chave, entidade e sentimento. Para obter mais informações, consulte Processamento assíncrono em lotes.	27 de junho de 2018
Novo guia	Esta é a primeira versão do Guia do desenvolvedor do Amazon Comprehend.	29 de novembro de 2017

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.