Amazon Comprehend Detect PII

AWS AI Service Cards

AWS AI Service Cards: Amazon Comprehend Detect PII

Table of Contents

Amazon Comprehend Detect PII 1
Overview 1
Intended use cases and limitations 2
Design of Comprehend Detect PII 3
Deployment and performance optimization best practices 6
Further information 7
Glossary 7

Amazon Comprehend Detect PII

An AWS AI Service Card explains the use cases for which the service is intended, how machine learning (ML) is used by the service, and key considerations in the responsible design and use of the service. A Service Card will evolve as AWS receives customer feedback, and as the service progresses through its lifecycle. AWS recommends that customers assess the performance of any AI service on their own content for each use case they need to solve. For more information, please see AWS Responsible Use of AI Guide and the references at the end. Please also be sure to review the AWS Responsible AI Policy, AWS Acceptable Use Policy, and AWS Service Terms for the services you plan to use.

This Service Card applies to the release of Comprehend Detect PII that is current as of November 11, 2023.

Overview

Amazon Comprehend Detect PII is designed to enable customers to detect PII entities in English text documents. A PII entity is a specific type of personally identifiable information (PII), which is data that can be used to identify an individual, such as a name, phone number or bank account number. This AI Service Card describes considerations for responsibly using the Detect PII API to identify PII in text in real time or in asynchronous analysis jobs. Typically, customers use this feature to locate PII data or to redact (mask or hide) PII data within text documents. Redacting PII helps the customer protect the privacy of their end-users and other individuals, hide sensitive data and control access to PII data per user role.

Comprehend Detect PII inspects text for text patterns called entities. An entity could be a person, place, or organization, as well as dates and identification numbers. From the list of all entities in a given sentence, Detect PII identifies the subset of entities containing PII, such as a person's name, credit card number, or social security number. For example, in the sentence "Dear Paulo, please email your feedback for AnyCafe at anycafe@mail.com." the entities are "Paulo" (Person name), "AnyCafe" (Organization name), and "anycafe@mail.com" (Other), while the PII entities are "Paulo" (Name) and "anycafe@mail.com" (Email). For each PII entity, Detect PII provides the entity type, the entity position (where text begins and ends), and a confidence score (0.0 to 1.0) which is the likelihood that the entity contains PII. In the above example, the entity "anycafe@mail.com" results in entity type ("Email"), located in text offset (begin 54, end 70) and with a confidence score (0.99). To evaluate the accuracy of Detect PII, Comprehend compares the service predictions to results generated by human experts. Each detected PII entity is a hit if the prediction matches

annotated results, and a miss otherwise. To evaluate the accuracy of the feature we use accuracy metrics including precision, recall, and F1 score; F1 is a balanced score between the precision and recall, where a prediction is considered correct if it matches both the entity type and start/end of the entity.

Detect PII identifies two types of PII entities, (i) universal PII entities and (ii) country-specific PII entities. Universal PII entities are commonly used entities that are not locale-specific such as name, age, email address, and also system-generated entities such as IP address, MAC address, and aws_access_key. Country specific entities are locale specific entities such as U.S. social security number, U.K. taxpayer reference number, and other government issued ID numbers. Detect PII covers a fixed set of 22 universal and 14 country-specific PII entity types. For each of the PII entity types, cues from the text may vary widely. Entities like phone, credit card, or postal code might appear with different numbers of digits, or appear with/without separators (white-space, hyphen, plus symbol). The format of names can vary from country to country, such as first-name before last-name and vice-versa, single word names, names separated by hyphens, names with multiple middle names, etc. This intrinsic variation can make PII entity detection challenging. Errors in the source text can also cause confounding variations. For example, misspellings and typos can appear in a plain text source document, and text transcribed from speech may include speech recognition errors. Detect PII is trained on a wide variety of examples to recognize PII in text documents, which helps minimize impacts from the confounding and intrinsic variations.

Intended use cases and limitations

Comprehend Detect PII is intended for use on transcribed text data and non-transcribed plain-text data in the following English locales: en-US, en-GB, en-CA, and en-IN. The service does not support text from databases, native PDFs or languages other than English. Detect PII is trained to locate PII entities listed in Comprehend Detect PII taxonomy (22 universal and 14 PII entities across 4 countries U.S., U.K., Canada and India) and does not support custom PII entities (such as a hotel booking number). To identify custom PII entities, customers can use the custom entity recognition feature within the Comprehend service.

Customers can leverage Comprehend Detect PII's automatic PII detection and redaction capabilities to accelerate PII filtering within applications, manage data access by user-role, protect the privacy of individuals and help safeguard against data breaches. The appropriateness of using Detect PII depends on a number of factors, including the type of PII entities involved (e.g. name vs bank account number), associated responsible AI concerns (e.g. privacy, fairness), the relative costs of false positives (service tags a non-PII entity as PII) and false negatives (service does not tag a PII entity as PII), expected confounding variation, and other factors. Based on the use case, customers

can tune the service configuration (confidence threshold) to prioritize minimizing either favor false positives or false negatives. We organize Comprehend Detect PII applications into two broad use cases.

- **PII Redaction use case:** These are applications running in offline (asynchronous) mode that use Detect PII to hide detected PII entities within text to enable safer downstream processing, e.g., an insurance application that excludes customer PII entities such as names and bank account numbers from appearing in documents that must be shared as inputs to downstream applications. Redaction is not supported in real-time mode. In this use case, the cost of false negatives (not hiding a valid customer name, phone or bank account number) is very high compared to cost of false positives (hiding entities that are not customer name, phone or bank account number). By lowering the service confidence threshold, customers may reduce false negatives and capture more text span that have a potential of being a PII entity.

- **PII Location use case:** These are applications running in either real-time (synchronous) or offline (asynchronous) mode that use Detect PII to locate PII entities within text to enable downstream processing, e.g., a marketing application that extracts customer names and phone numbers from survey forms to store in a customer database. In this use case, the cost of false negatives (e.g., missing a name or phone number) may be lower in this use case compared to the above redaction use case (on hiding bank account numbers in insurance documents). By increasing the confidence threshold, customers can set the service to capture text with a higher precision (i.e., capture text with a higher probability of being PII).

Comprehend Detect PII performance results may vary with the evaluation dataset used to generate the results. We recommend that customers assess the service on their own content (data, settings and context types) to determine if a particular use case is suitable for the Detect PII API. Furthermore, for each use case, customers should assess the level of RAI risk and the related cost of false negatives/false positives and adjust the confidence threshold for predicting PII entities. Customers should also assess the need for human oversight and support the review of Detect PII output by human reviewers as needed.

Design of Comprehend Detect PII

Machine learning

Detect PII is built using ML and natural language processing technologies. It works as follows: Detect PII takes an English text document as input. An NLP model identifies text spans within

the document that belong to a range of PII elements and returns the text span identified as a PII entity with entity type, entity position (start, end) and a confidence score which is the probability of the text span being a PII entity of given PII entity type. For more information, see Personally identifiable information (PII) in the *Amazon Comprehend Developer Guide*.

Performance expectations

Confounding variation will differ between customer applications. As a result, the performance will vary between applications. For example, consider two different sets of support ticket applications A and B. Application A contains plain text files, and Application B contains text files generated by a native PDF to text converter. Documents in A are superior quality with well-formatted text files, whereas, documents in B may have formatting or other errors due to optical character recognition (OCR). Because of the possible differences in the quality of input text for A and B, the error rates will likely be different in detecting PII, assuming Comprehend is deployed perfectly in each application.

Test-driven methodology

To evaluate the performance of Comprehend Detect PII, we use multiple datasets of text documents containing PII data that vary based on demographic composition (volume and diversity), amount of confounding variation, types of labels and other factors. The overall performance on a given dataset is represented by an F1 score which ranges from 0.0 to 1.0. The best possible performance will score 1, whereas 0 will indicate the worst. F1 score provides a tradeoff between the model's precision (percentage of predicted fields that are correct) and recall (percentage of correct fields that are included in the prediction). Changing the confidence threshold of the Detect PII model can alter the precision and recall metrics. Groups in a dataset can be defined by key attributes (such as gender and country), confounding variables (such as original text or OCR extracted text), or a combination of the two. Because of these variations, the F1 scores, both overall and for groups, differ from dataset to dataset. Taking this into consideration, our development process iteratively examines the performance on multiple datasets, attempts to increase F1 scores for groups on which Detect PII did not perform as well and then uses it to improve the suite of evaluation datasets.

Fairness and bias

Our goal is for Detect PII to identify PII entities from text documents with high accuracy irrespective of the demographic and geo-specific attributes of the PII entity represented in the document. To achieve this, we use the iterative development process, including building training datasets to capture a range of locales and documents supported by Detect PII, under a range of text quality conditions. We routinely test on datasets of text files for which we

have reliable PII entity labels. Detect PII performs consistently well across many locales and demographic attributes. For example, on an internal dataset that we tested, we found that the F1 score of detecting names from 26 different locales across the world did not drop below 87%; the F1 for detecting phone numbers and addresses across the 4 locales supported by Detect PII did not go below 88% and 91% respectively. Because results depend not only on Detect PII but also on the customer workflow and the evaluation dataset, we recommend that customers test Detect PII on their own content.

Explainability

If customers have questions regarding the prediction returned by Comprehend for a specific PII entity, we suggest that customers use the entity, entity start and end span and confidence score returned by Comprehend to directly examine the PII entities in the given text documents.

Robustness

We maximize robustness with a number of techniques, including using large training datasets that capture many kinds of variation across many documents. As part of this process, we build datasets to capture the range of variations to detect PII, under a range of conditions for text quality. Inputs that are easiest for Detect PII to process contain text that are relatively free from spelling errors, white space errors, special characters, and other formatting issues. However, Detect PII models are trained to be resilient even when inputs vary from ideal conditions.

Privacy and security

Detect PII captures and processes text. Inputs and outputs are never shared between customers. Comprehend Detect PII does not train on customer content. For more information, see Section 50.3 of the AWS Service Terms and the AWS Data Privacy FAQs. For service-specific privacy and security information, see Amazon Comprehend FAQs and Amazon Comprehend Security.

Transparency

Where appropriate for their use case, customers who incorporate Detect PII into their workflow should consider disclosing their use of ML to end users and other individuals impacted by the application, and give their end users the ability to provide feedback to improve workflows. In their documentation, customers can also reference this AI Service Card.

Governance

We have rigorous methodologies to build our AWS AI services responsibly, including a working backwards product development process that incorporates Responsible AI at the design phase, in design consultations, and implementation assessments by dedicated Responsible AI science

and data experts, routine testing, reviews with customers, and best practice development, dissemination, and training.

Deployment and performance optimization best practices

We encourage customers to build and operate their applications responsibly, as described in the AWS Responsible Use of AI Guide. This includes implementing responsible AI practices to address key dimensions including fairness and bias, robustness, explainability, privacy and security, transparency, and governance.

Workflow Design: The performance of any application using Comprehend Detect PII depends on the design of the customer workflow, including: (1) the amount of confounding variation, (2) intrinsic variation, (3) selection of confidence thresholds, (4) human oversight, (5) how consistently the workflow is applied across demographic groups, and (6) periodic retesting for performance drift.

1. **Confounding variation**: When working with text documents from sources such as OCR output and transcribed text, it is important to incorporate steps into the workflow that reduce spelling, formatting and other kinds of preprocessing errors. In cases where there is significant confounding variation, consider including quality assurance steps, encompassing the expected range of variations, including transcription or OCR quality. For yielding best results from Detect PII, customer workflows should define policies regarding acceptable input text and regularly assess compliance by randomly and periodically sampling inputs.
2. **Intrinsic variation**: When working with text documents with entities consisting of multiple words, it is important to incorporate steps into the workflows that aim to handle entities of varying formats and lengths. For example, an address in text may show up as a street number and street address and can optionally include city, state and country which can result in an address being predicted as two different entities. To handle this situation, customers can consider post-processing steps to merge Comprehend PII's prediction output. Alternatively, customers can consider using custom entity recognition.
3. **Confidence thresholding**: Customers may tune performance by adjusting the confidence threshold setting (0.0 to 1.0) on the service for detecting PII entities within a text. Based on the confidence threshold, the service identifies entities with a greater or smaller probability. For better precision, choose a high threshold. For better recall, choose a lower threshold. To set an appropriate threshold, a customer may collect a representative set of inputs, label the text fields of each, and try higher or lower thresholds until satisfied with the user experience.

4. **Human oversight:** If a customer's application workflow involves a high risk or sensitive use case, such as a decision that impacts an individual's rights or access to essential services, human review should be incorporated into the application workflow where appropriate. Automatic entity recognition with Detect PII can serve as a tool to reduce the effort incurred by fully manual solutions, and to allow humans to expeditiously review and assess sensitive documents containing PII entities.
5. **Consistency**: Customers should set and enforce policies for the kinds of input text permitted, and for how humans combine the use of confidence thresholding and their own judgment to determine final results. These policies should be consistent across demographic groups.
6. **Performance drift**: Changes in the kind of text that a customer submits to Detect PII, or changes to the service, may lead to different outputs. To address these changes, customers should consider periodically retesting the performance of Detect PII, and adjusting their workflow if necessary.

Further information

- For service documentation, see Amazon Comprehend Developer Guide.
- For examples of detecting and masking PII workflows, see Detecting and redacting PII using Amazon Comprehend, Redacting PII from application log output with Amazon Comprehend, Redact sensitive data from streaming data in near-real time using Amazon Comprehend and Amazon Data Firehose.
- To learn more about quality metrics such as Precision, Recall and F1 score, see Automatically generate model evaluation metrics using SageMaker AI Autopilot Model Quality Reports.
- For details on privacy and other legal considerations, see the following AWS policies: Acceptable Use, Responsible AI, Legal, Compliance, and Privacy.
- For help optimizing workflows, see Generative AI Innovation Center, AWS Customer Support, AWS Professional Services, Ground Truth Plus, and Amazon Augmented AI.
- If you have any questions or feedback about AWS AI service cards, please complete this form.

Glossary

Controllability: Steering and monitoring AI system behavior.

Privacy & Security: Appropriately obtaining, using and protecting data and models.

Safety: Preventing harmful system output and misuse.

Fairness: Considering impacts on different groups of stakeholders.

Explainability: Understanding and evaluating system outputs.

Veracity & Robustness: Achieving correct system outputs, even with unexpected or adversarial inputs.

Transparency: Enabling stakeholders to make informed choices about their engagement with an AI system.

Governance: Incorporating best practices into the AI supply chain, including providers and deployers.