



Guida per gli sviluppatori

Amazon Comprehend



Amazon Comprehend: Guida per gli sviluppatori

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è Amazon Comprehend?	1
Informazioni dettagliate su Amazon Comprehend	2
Amazon Comprehend personalizzato	2
Volani	3
Raggruppamento di documenti (modellazione di argomenti)	3
Esempi	3
Vantaggi	4
Prezzi di Amazon Comprehend	4
Sei un utente alle prime armi di Amazon Comprehend?	5
Come funziona	6
Approfondimenti	6
Entità	7
Eventi	9
Frase chiave	17
Lingua dominante	18
Sentiment	25
Sentimento mirato	26
Analisi della sintassi	42
Amazon Comprehend personalizzato	47
Modellazione degli argomenti	48
Modalità di elaborazione dei documenti	52
Elaborazione di un singolo documento	52
Elaborazione sincrona di più documenti	52
Elaborazione asincrona in batch	55
Lingue supportate	57
Lingue supportate	57
Lingue supportate dalle funzionalità di Amazon Comprehend	58
Configurazione	60
Iscriviti per un Account AWS	60
Crea un utente con accesso amministrativo	60
Configura il AWS CLI	62
Concessione dell'accesso programmatico	62
Nozioni di base	65
Utilizzo della console	67

Analisi in tempo reale	67
Entità	68
Frase chiave	69
Lingua	70
Informazioni personali di identificazione (PII)	71
Sentiment	73
Sentimento mirato	74
Sintassi	76
Lavori di analisi (console)	77
Utilizzo dell'API	81
Lavorare con AWS SDKs	81
Analisi in tempo reale (API)	82
Rilevare la lingua dominante	83
Rilevamento di entità denominate	84
Rilevamento delle frasi chiave	85
Determinare il sentimento	86
Analisi in tempo reale per un sentiment mirato	88
Rilevamento della sintassi	90
Batch in tempo reale APIs	92
Lavori di analisi asincrona (API)	97
Informazioni dettagliate su Amazon Comprehend	98
Sentimento mirato	104
Rilevamento degli eventi	105
Modellazione degli argomenti	110
Fiducia e sicurezza	114
Rilevamento della tossicità	115
Rilevamento di contenuti tossici tramite l'API	116
Classificazione rapida della sicurezza	118
Classificazione rapida della sicurezza tramite l'API	119
Rilevamento e redazione delle PII	121
Informazioni personali di identificazione (PII)	122
Rilevamento di entità PII	122
Individua le entità PII	123
Redigi le entità PII	124
Tipi di entità universali PII	124
Tipi di entità PII specifici per paese	127

Etichettatura delle entità PII	129
Analisi in tempo reale (console)	130
Offset	71
Etichette	73
Lavori di analisi asincrona (Console)	132
Analisi in tempo reale (API)	134
Individuazione delle entità PII in tempo reale (API)	135
Etichettatura delle entità PII in tempo reale (API)	136
Lavori di analisi asincrona (API)	137
Individuazione delle entità PII	137
Redazione delle entità PII	142
Elaborazione dei documenti	147
Ingressi per analisi in tempo reale	147
Documenti in testo semplice	148
Documenti semistrutturati	148
File di immagine e file PDF scansionati	148
Output di Amazon Textract	148
Dimensioni massime dei documenti per l'analisi in tempo reale	148
Errori nei documenti semistrutturati	149
Ingressi per l'analisi asincrona	150
Documenti in testo semplice	150
Documenti semistrutturati	151
File di immagine e file PDF scansionati	152
File JSON di output di Amazon Textract	152
Impostazione delle opzioni di estrazione del testo	153
Procedure consigliate per le immagini	154
Classificazione personalizzata	155
Preparazione dei dati di allenamento	155
Formati di file di addestramento	156
Modalità multiclasse	158
Modalità multi-etichetta	161
Modelli di classificazione della formazione	164
Addestra classificatori personalizzati (console)	165
Addestra classificatori personalizzati (API)	170
Verifica i dati di allenamento	172
Risultati formativi di Classifier	173

Metriche	178
Analisi in corso in tempo reale	183
Analisi in tempo reale (console)	184
Analisi in tempo reale (API)	186
Uscite per analisi in tempo reale	189
Esecuzione di processi di analisi asincrona	191
Formati di file di input	191
Lavori di analisi (console)	193
Lavori di analisi (API)	195
Output per lavori di analisi	197
Riconoscimento personalizzato delle entità	201
Preparazione dei dati di addestramento	202
Quando utilizzare le annotazioni rispetto agli elenchi di entità	203
Elenchi di entità	204
Annotazioni	206
Modelli di riconoscimento della formazione	220
Addestra riconoscitori personalizzati (console)	221
Addestra riconoscitori personalizzati (API)	228
Metriche	230
Eseguire analisi in tempo reale	234
Analisi in tempo reale (console)	235
Analisi in tempo reale (API)	237
Uscite per analisi in tempo reale	239
Esecuzione di processi di analisi asincrona	245
Lavori di analisi (console)	246
Lavori di analisi (API)	248
Output per lavori di analisi	252
Gestione di modelli personalizzati	257
Controllo delle versioni dei modelli con Amazon Comprehend	257
Copiare modelli personalizzati tra Account AWS	260
Condivisione di un modello personalizzato	261
Importazione di un modello personalizzato	270
Volani	278
Panoramica del volano	278
Set di dati Flywheel	279
Creazione del volano	280

Stati del volano	280
Iterazioni Flywheel	281
Data lake Flywheel	281
Struttura delle cartelle Data Lake	282
Gestione del data lake	283
Policy e autorizzazioni IAM	284
Configura le autorizzazioni utente IAM	284
Configura le autorizzazioni per le chiavi AWS KMS	285
Creare un ruolo di accesso ai dati	285
Configurazione dei volani (console)	285
Create un volano	285
Aggiorna un volano	288
Eliminare un volano	288
Configurazione dei volani (API)	289
Create un volano per un modello esistente	289
Crea un volano per un nuovo modello	289
Descrivi un volano	290
Aggiorna un volano	291
Eliminare un volano	291
Elenca i volani	292
Configurazione dei set di dati	292
Creazione di un set di dati (console)	293
Creazione di un set di dati (API)	293
Descrivi un set di dati	294
Iterazioni del volano	294
Flusso di lavoro di iterazione	295
Gestione delle iterazioni (console)	295
Gestione delle iterazioni (API)	296
Utilizzo dei volani	299
Analisi in tempo reale	299
Lavori asincroni	300
Gestione degli endpoint	301
Panoramica degli endpoint	301
Utilizzo degli endpoint	302
Monitoraggio degli endpoint	303
Aggiornamento degli endpoint	306

Usando Trusted Advisor	308
Endpoint Amazon Comprehend sottoutilizzati	308
Rischio di accesso agli endpoint di Amazon Comprehend	310
Eliminazione di endpoint	312
Scalabilità automatica con endpoint	313
Monitoraggio degli obiettivi	314
Dimensionamento pianificato	319
Assegnazione di tag	323
Taggare una nuova risorsa	324
Visualizzazione, modifica ed eliminazione dei tag	325
Esempi di codice	327
Nozioni di base	328
Azioni	329
Scenari	412
Creazione di un'app in streaming Amazon Transcribe	413
Creazione di un chatbot Amazon Lex	413
Creare un'applicazione di messaggistica	415
Crea un'applicazione per analizzare il feedback dei clienti	416
Rilevare gli elementi del documento	422
Rilevamento di entità nel testo estratto da un'immagine	428
Eseguire un processo di modellazione di argomenti in base a dati di esempio	428
Addestrare un classificatore personalizzato e classificare i documenti	433
Sicurezza	447
Protezione dei dati	448
Crittografia KMS in Amazon Comprehend	449
Prevenzione del confused deputy tra servizi	452
Utilizzo di un Virtual Private Cloud (VPC)	455
Endpoint VPC (AWS PrivateLink)	462
Identity and Access Management	463
Destinatari	464
Autenticazione con identità	464
Gestione dell'accesso tramite policy	466
Come funziona Amazon Comprehend con IAM	468
Esempi di policy basate su identità	474
AWS politiche gestite	487
Risoluzione dei problemi	491

Registrazione delle chiamate API Amazon Comprehend con AWS CloudTrail	493
Amazon Comprehend: informazioni in CloudTrail	493
Esempio: voci dei file di log di Amazon Comprehend	496
Convalida della conformità	498
Resilienza	499
Sicurezza dell'infrastruttura	499
Linee guida e quote	500
Regioni supportate	500
Quote per i modelli integrati	501
Analisi (sincrona) in tempo reale	501
Analisi asincrona	503
Quote per modelli personalizzati	506
Quote generali	506
Quote per gli endpoint	506
Classificazione dei documenti	507
Riconoscimento personalizzato delle entità	512
Quote per i volani	516
Quote generali per i volani	516
Quote del set di dati per modelli di classificazione personalizzati	517
Quote del set di dati per modelli di riconoscimento delle entità personalizzati	517
Esercitazioni	518
Analisi delle informazioni ricavate dalle recensioni	518
Prerequisiti	520
Fase 1: aggiunta di documenti ad Amazon S3	522
Fase 2: (solo CLI) creazione di un ruolo IAM	526
Fase 3: Esecuzione di processi di analisi	530
Fase 4: Preparazione dell'output	534
Fase 5: Visualizzazione dell'output	546
Utilizzo dei punti di accesso Lambda a oggetti S3 per PII	552
Controllo dell'accesso ai documenti con PII	553
Eliminazione delle PII dai documenti	555
Analisi del testo con OpenSearch	557
Guida di riferimento alle API	558
Cronologia dei documenti	559
AWS Glossario	575
.....	dlxxvi

Che cos'è Amazon Comprehend?

Amazon Comprehend utilizza l'elaborazione del linguaggio naturale (NLP) per estrarre informazioni dettagliate sul contenuto dei documenti. Sviluppa informazioni basandosi su entità, frasi chiave, linguaggio, sentimenti e altri elementi comuni in un documento. Usa Amazon Comprehend per creare nuovi prodotti basati sulla comprensione della struttura dei documenti. Ad esempio, con Amazon Comprehend puoi cercare nei feed dei social network le menzioni di prodotti o scansionare un intero archivio di documenti alla ricerca di frasi chiave.

Puoi accedere alle funzionalità di analisi dei documenti di Amazon Comprehend utilizzando la console Amazon Comprehend o Amazon Comprehend. APIs Puoi eseguire analisi in tempo reale per piccoli carichi di lavoro o avviare lavori di analisi asincroni per set di documenti di grandi dimensioni. Puoi utilizzare i modelli preaddestrati forniti da Amazon Comprehend oppure puoi addestrare i tuoi modelli personalizzati per la classificazione e il riconoscimento delle entità.

Amazon Comprehend può archiviare i tuoi contenuti per migliorare continuamente la qualità dei suoi modelli preformati. Per ulteriori informazioni, consulta le [domande frequenti su Amazon Comprehend](#).

Tutte le funzionalità di Amazon Comprehend accettano documenti di testo UTF-8 come input. Inoltre, la classificazione personalizzata e il riconoscimento personalizzato delle entità accettano file di immagine, file PDF e file Word come input.

Amazon Comprehend può esaminare e analizzare documenti in diverse lingue, a seconda della funzionalità specifica. Per ulteriori informazioni, consulta [Lingue supportate in Amazon Comprehend](#). La [Lingua dominante](#) funzionalità di Amazon Comprehend può esaminare documenti e determinare la lingua dominante per una selezione molto più ampia di lingue.

Argomenti

- [Informazioni dettagliate su Amazon Comprehend](#)
- [Amazon Comprehend personalizzato](#)
- [Volani](#)
- [Raggruppamento di documenti \(modellazione di argomenti\)](#)
- [Esempi](#)
- [Vantaggi](#)
- [Prezzi di Amazon Comprehend](#)
- [Sei un utente alle prime armi di Amazon Comprehend?](#)

Informazioni dettagliate su Amazon Comprehend

Amazon Comprehend utilizza un modello pre-addestrato per esaminare e analizzare un documento o un set di documenti per raccogliere informazioni al riguardo. Questo modello viene continuamente addestrato su un ampio corpus di testo in modo che non sia necessario fornire dati di addestramento.

Amazon Comprehend analizza i seguenti tipi di informazioni:

- **Entità:** riferimenti ai nomi di persone, luoghi, oggetti e luoghi contenuti in un documento.
- **Frase chiave:** frasi che compaiono in un documento. Ad esempio, un documento su una partita di basket potrebbe riportare i nomi delle squadre, il nome della sede e il risultato finale.
- **Informazioni di identificazione personale (PII):** dati personali che possono identificare un individuo, come un indirizzo, un numero di conto bancario o un numero di telefono.
- **Lingua:** la lingua dominante di un documento.
- **Sentimento:** il sentimento dominante di un documento, che può essere positivo, neutro, negativo o misto.
- **Sentimento mirato:** i sentimenti associati a entità specifiche in un documento. Il sentimento relativo alla presenza di ciascuna entità può essere positivo, negativo, neutro o misto.
- **Sintassi:** le parti del discorso per ogni parola del documento.

Per ulteriori informazioni, consulta [Approfondimenti](#).

Amazon Comprehend personalizzato

Puoi personalizzare Amazon Comprehend in base ai tuoi requisiti specifici senza le competenze necessarie per creare soluzioni NLP basate sull'apprendimento automatico. Utilizzando l'apprendimento automatico, o AutoML, Amazon Comprehend Custom crea modelli NLP personalizzati per tuo conto, utilizzando i dati che già possiedi.

Classificazione personalizzata: crea modelli di classificazione personalizzati (classificatori) per organizzare i documenti nelle tue categorie.

Riconoscimento personalizzato delle entità: crea modelli di riconoscimento delle entità personalizzati (riconoscitori) in grado di analizzare il testo in base a termini e frasi sostantivi specifici.

Per ulteriori informazioni, consulta [Amazon Comprehend personalizzato](#).

Volani

Utilizza i volani per semplificare il processo di addestramento e gestione delle versioni personalizzate dei modelli nel tempo. Un volano aiuta a orchestrare le attività associate all'addestramento e alla valutazione di nuove versioni di un modello. Flywheels supporta modelli personalizzati in testo semplice per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Per ulteriori informazioni, consulta [Volani](#).

Raggruppamento di documenti (modellazione di argomenti)

Puoi anche usare Amazon Comprehend per esaminare un corpus di documenti e organizzarli in base a parole chiave simili al loro interno. Il raggruppamento di documenti (modellazione degli argomenti) è utile per organizzare un ampio corpus di documenti in argomenti o cluster simili in base alla frequenza delle parole. Per ulteriori informazioni, consulta [Modellazione degli argomenti](#).

Esempi

Gli esempi seguenti mostrano come utilizzare le operazioni di Amazon Comprehend nelle tue applicazioni.

Example 1: Trova documenti su un argomento

Trova i documenti su un argomento specifico utilizzando la modellazione tematica di Amazon Comprehend. Scansiona un set di documenti per determinare gli argomenti discussi e per trovare i documenti associati a ciascun argomento. Puoi specificare il numero di argomenti che Amazon Comprehend deve restituire dal set di documenti.

Example 2: Scopri cosa pensano i clienti dei tuoi prodotti

Se la tua azienda pubblica un catalogo, lascia che Amazon Comprehend ti dica cosa pensano i clienti dei tuoi prodotti. Invia a ogni cliente un commento sull'`DetectSentiment` operazione e ti dirà se i clienti si sentono positivi, negativi, neutrali o confusi riguardo a un prodotto.

Example 3: Scopri cosa è importante per i tuoi clienti

Usa la modellazione di argomenti di Amazon Comprehend per scoprire gli argomenti di cui parlano i tuoi clienti nei forum e nelle bacheche, quindi utilizza il rilevamento delle entità per determinare le

persone, i luoghi e le cose che associano all'argomento. Usa l'analisi del sentiment per determinare l'opinione dei tuoi clienti su un argomento.

Vantaggi

I vantaggi dell'utilizzo di Amazon Comprehend includono:

- **Integra una potente elaborazione del linguaggio naturale nelle tue app:** Amazon Comprehend rimuove la complessità legata all'integrazione di funzionalità di analisi del testo nelle tue applicazioni rendendo disponibile un'elaborazione del linguaggio naturale potente e accurata con una semplice API. Non hai bisogno di esperienza nell'analisi testuale per sfruttare le informazioni fornite da Amazon Comprehend.
- **Elaborazione del linguaggio naturale basata sul deep learning:** Amazon Comprehend utilizza la tecnologia di deep learning per analizzare accuratamente il testo. I nostri modelli vengono costantemente addestrati con nuovi dati su più domini per migliorare la precisione.
- **Elaborazione scalabile del linguaggio naturale:** Amazon Comprehend ti consente di analizzare milioni di documenti per scoprire le informazioni in essi contenute.
- **Integrato con altri AWS servizi:** Amazon Comprehend è progettato per funzionare perfettamente con altri AWS servizi come Amazon S3 e AWS KMS. AWS Lambda Archivia i tuoi documenti in Amazon S3 o analizza i dati in tempo reale con Firehose. Support for AWS Identity and Access Management (IAM) semplifica il controllo sicuro dell'accesso alle operazioni di Amazon Comprehend. Con IAM, puoi creare e gestire utenti e gruppi per concedere l'accesso appropriato ai tuoi sviluppatori e utenti finali.
- **Crittografia dei risultati di output e dei dati di volume:** Amazon S3 consente già di crittografare i documenti di input e Amazon Comprehend estende ulteriormente questa funzionalità. Utilizzando la tua chiave KMS, puoi crittografare i risultati di output del tuo lavoro e i dati sul volume di storage collegato all'istanza di calcolo che elabora il processo di analisi. Il risultato è una sicurezza notevolmente migliorata.
- **Basso costo:** con Amazon Comprehend, non sono previste commissioni minime o impegni iniziali. Paghi per i documenti che analizzi e per i modelli personalizzati che addestra.

Prezzi di Amazon Comprehend

Con Amazon Comprehend, paghi solo per le risorse che utilizzi. Se sei un nuovo AWS cliente, puoi iniziare a usare Amazon Comprehend gratuitamente. Per ulteriori informazioni, consulta il [piano AWS di utilizzo gratuito](#).

È previsto un costo di utilizzo per l'esecuzione di lavori di analisi in tempo reale o asincroni. Paghi per addestrare modelli personalizzati e paghi per la gestione dei modelli personalizzati. Per le richieste in tempo reale che utilizzano modelli personalizzati, si paga l'endpoint dal momento in cui si avvia l'endpoint fino all'eliminazione dell'endpoint. Non sono previsti costi aggiuntivi per l'utilizzo dei volani. Tuttavia, quando si esegue un'iterazione del volano, si devono sostenere i costi standard per l'addestramento di una nuova versione del modello e l'archiviazione dei dati del modello.

Per le tariffe e ulteriori informazioni dettagliate, consulta la pagina dei prezzi di [Amazon Comprehend](#).

Sei un utente alle prime armi di Amazon Comprehend?

Se sei un utente alle prime armi di Amazon Comprehend, ti consigliamo di leggere le seguenti sezioni nell'ordine:

1. [Come funziona](#)— Questa sezione introduce i concetti di Amazon Comprehend.
2. [Configurazione](#)— In questa sezione, crei un account e configuri il. AWS CLI
3. [Guida introduttiva ad Amazon Comprehend](#)— In questa sezione, esegui un processo di analisi di Amazon Comprehend.
4. [Tutorial: analisi delle informazioni ricavate dalle recensioni dei clienti con Amazon Comprehend](#)— In questa sezione, esegui l'analisi del sentimento e delle entità e visualizzi i risultati.
5. Riferimento all'[API Amazon Comprehend: documentazione di riferimento](#) per le operazioni di Amazon Comprehend.

AWS fornisce le seguenti risorse per conoscere il servizio Amazon Comprehend:

- Il [AWS Machine Learning Blog](#) include articoli utili su Amazon Comprehend.
- [Amazon Comprehend Resources](#) fornisce video e tutorial utili su Amazon Comprehend.

Come funziona

Amazon Comprehend utilizza un modello pre-addestrato per raccogliere informazioni su un documento o un set di documenti. Questo modello viene continuamente addestrato su un ampio corpo di testo in modo che non sia necessario fornire dati di addestramento.

Puoi usare Amazon Comprehend per creare modelli personalizzati per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Puoi usarlo [Volani](#) per aiutare a gestire i modelli personalizzati.

Amazon Comprehend fornisce modelli di argomenti utilizzando un modello integrato. La modellazione tematica esamina un corpus di documenti e organizza i documenti in base a parole chiave simili al loro interno.

Amazon Comprehend fornisce modalità di elaborazione dei documenti sincrone e asincrone. Utilizza la modalità sincrona per elaborare un documento o un batch di massimo 25 documenti. Utilizzate un processo asincrono per elaborare un gran numero di documenti.

Amazon Comprehend collabora con AWS Key Management Service (AWS KMS) per fornire una crittografia avanzata per i tuoi dati. Per ulteriori informazioni, consulta [Crittografia KMS in Amazon Comprehend](#).

Concetti chiave

- [Approfondimenti](#)
- [Amazon Comprehend personalizzato](#)
- [Modellazione degli argomenti](#)
- [Modalità di elaborazione dei documenti](#)

Approfondimenti

Amazon Comprehend può analizzare un documento o un set di documenti per raccogliere informazioni al riguardo. Alcune delle informazioni che Amazon Comprehend sviluppa su un documento includono:

- [Entità](#)— Amazon Comprehend restituisce un elenco di entità, come persone, luoghi e località, identificate in un documento.

- [Eventi](#)— Amazon Comprehend rileva tipi specifici di eventi e dettagli correlati.
- [Frase chiave](#)— Amazon Comprehend estrae le frasi chiave che compaiono in un documento. Ad esempio, un documento su una partita di basket potrebbe riportare i nomi delle squadre, il nome della sede e il punteggio finale.
- [Informazioni personali di identificazione \(PII\)](#)— Amazon Comprehend analizza i documenti per rilevare dati personali che identificano un individuo, come indirizzo, numero di conto corrente bancario o numero di telefono.
- [Lingua dominante](#)— Amazon Comprehend identifica la lingua dominante in un documento. Amazon Comprehend è in grado di identificare 100 lingue.
- [Sentimento](#): Amazon Comprehend determina il sentimento dominante di un documento. Il sentimento può essere positivo, neutro, negativo o misto.
- [Sentimento mirato](#): Amazon Comprehend determina il sentimento di entità specifiche menzionate in un documento. Il sentimento di ogni menzione può essere positivo, neutro, negativo o misto.
- [Analisi della sintassi](#)— Amazon Comprehend analizza ogni parola del documento e determina la parte del discorso corrispondente. Ad esempio, nella frase «Sta piovendo oggi a Seattle», «it» viene identificato come pronome, «raining» è identificato come un verbo e «Seattle» è identificato come un sostantivo proprio.

Entità

Un'entità è un riferimento testuale al nome univoco di un oggetto del mondo reale come persone, luoghi e articoli commerciali e a riferimenti precisi a misure come date e quantità.

Ad esempio, nel testo «John si è trasferito al 1313 di Mockingbird Lane nel 2012», «John» potrebbe essere riconosciuto come unPERSON, «1313 Mockingbird Lane» potrebbe essere riconosciuto come unLOCATION e «2012» potrebbe essere riconosciuto come unDATE.

Ogni entità ha anche un punteggio che indica il livello di fiducia di Amazon Comprehend rispetto al corretto rilevamento del tipo di entità. Puoi filtrare le entità con punteggi più bassi per ridurre il rischio di utilizzare rilevamenti errati.

La tabella seguente elenca i tipi di entità.

Tipo	Description
ARTICOLO_ COMMERCIALE	Un prodotto di marca
DATE	Una data completa (ad esempio, 25/11/2017), giorno (martedì), mese (maggio) o ora (8:30)
EVENT	Un evento, come un festival, un concerto, un'elezione, ecc.
LOCATION	Un luogo specifico, ad esempio un paese, una città, un lago, un edificio, ecc.
ORGANIZZAZIONE	Grandi organizzazioni, come un governo, un'azienda, una religione, una squadra sportiva, ecc.
OTHER	Entità che non rientrano in nessuna delle altre categorie di entità
PERSONA	Individui, gruppi di persone, soprannomi, personaggi immaginari
QUANTITÀ	Un importo quantificato, ad esempio valuta, percentuali, numeri, byte, ecc.
TITOLO	Un nome ufficiale dato a qualsiasi creazione o opera creativa, come film, libri, canzoni, ecc.

Le operazioni di rilevamento delle entità possono essere eseguite utilizzando una qualsiasi delle lingue principali supportate da Amazon Comprehend. Ciò include solo il rilevamento di entità predefinite (non personalizzate). Tutti i documenti devono essere nella stessa lingua.

È possibile utilizzare una delle seguenti operazioni API per rilevare le entità in un documento o in un set di documenti.

- [DetectEntities](#)
- [BatchDetectEntities](#)
- [StartEntitiesDetectionJob](#)

Le operazioni restituiscono un elenco di oggetti [API Entity](#), uno per ogni entità nel documento. L'BatchDetectEntitiesoperazione restituisce un elenco di Entity oggetti, un elenco per ogni documento del batch. L'StartEntitiesDetectionJoboperazione avvia un lavoro asincrono che produce un file contenente un elenco di Entity oggetti per ogni documento del lavoro.

L'esempio seguente è la risposta dell'operazione. 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"
}
```

Eventi

Usa il rilevamento degli eventi per analizzare i documenti di testo per tipi specifici di eventi e le relative entità. Amazon Comprehend supporta il rilevamento di eventi in grandi raccolte di documenti utilizzando processi di analisi asincroni. Per ulteriori informazioni sugli eventi, inclusi esempi di lavori di analisi degli eventi, consulta [Annuncio del lancio di Amazon Comprehend Events](#)

Entità

Dal testo di input, Amazon Comprehend estrae un elenco di entità correlate all'evento rilevato. Un'entità può essere un oggetto del mondo reale, come una persona, un luogo o un luogo; un'entità può anche essere un concetto, ad esempio una misurazione, una data o una quantità. Ogni occorrenza di un'entità è identificata da una menzione, che è un riferimento testuale all'entità nel testo di input. Per ogni entità unica, tutte le menzioni sono raggruppate in un elenco. Questo elenco

fornisce dettagli per ogni posizione nel testo di input in cui si trova l'entità. Amazon Comprehend rileva solo le entità associate ai tipi di eventi supportati.

Ogni entità associata a un tipo di evento supportato restituisce i seguenti dettagli correlati:

- **Menzioni:** dettagli per ogni occorrenza della stessa entità nel testo di input.
 - **BeginOffset:** Un offset di caratteri nel testo di input che mostra dove inizia la menzione (il primo carattere è nella posizione 0).
 - **EndOffset:** Un offset di caratteri nel testo di input che mostra dove finisce la menzione.
 - **Punteggio:** il livello di fiducia di Amazon Comprehend nell'accuratezza del tipo di entità.
 - **GroupScore:** Il livello di fiducia di Amazon Comprehend che la menzione sia correttamente raggruppata con altre menzioni della stessa entità.
 - **Testo:** il testo dell'entità.
 - **Tipo:** il tipo di entità. Per tutti i tipi di entità supportati, consulta [Tipi di entità](#).

Eventi

Amazon Comprehend restituisce l'elenco degli eventi (dei tipi di eventi supportati) che rileva nel testo di input. Ogni evento viene restituito con i seguenti dettagli correlati:

- **Tipo:** il tipo di evento. Per tutti i tipi di eventi supportati, consulta [Event types \(Tipi di evento\)](#).
- **Argomenti:** un elenco di argomenti correlati all'evento rilevato. Un argomento è costituito da un'entità correlata all'evento rilevato. Il ruolo dell'argomento descrive la relazione, ad esempio chi ha fatto cosa, dove e quando.
 - **EntityIndex:** un valore di indice che identifica un'entità dall'elenco di entità restituito da Amazon Comprehend per questa analisi.
 - **Ruolo:** il tipo di argomento, che descrive in che modo l'entità per questo argomento è correlata all'evento. Per tutti i tipi di argomenti supportati, vedere [Tipi di argomenti](#).
 - **Punteggio:** il livello di fiducia di Amazon Comprehend nell'accuratezza del rilevamento del ruolo.
- **Trigger:** un elenco di trigger per l'evento rilevato. Un trigger è una singola parola o frase che indica il verificarsi dell'evento.
 - **BeginOffset:** Un offset di caratteri nel testo di input che mostra dove inizia il trigger (il primo carattere si trova nella posizione 0).
 - **EndOffset:** Un offset di caratteri nel testo di input che mostra dove finisce il trigger.
 - **Punteggio:** il livello di fiducia di Amazon Comprehend nell'accuratezza del rilevamento.

- Testo: il testo del trigger.
- GroupScore: il livello di fiducia di Amazon Comprehend che il trigger sia correttamente raggruppato con altri trigger per lo stesso evento.
- Tipo: il tipo di evento indicato da questo trigger.

Rileva il formato dei risultati degli eventi

Una volta completato il processo di rilevamento degli eventi, Amazon Comprehend scrive i risultati dell'analisi nella posizione di output di Amazon S3 specificata all'avvio del processo.

Per ogni evento rilevato, l'output fornisce dettagli nel seguente 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"  
    }, ...  
]  
, ...  
]
```

Tipi supportati per entità, eventi e argomenti

Tipi di entità

Tipo	Description
DATE	Qualsiasi riferimento a una data o a un'ora, specifica o generale.
SERVIZIO, STRUTTURA	Edifici, aeroporti, autostrade, ponti e altre strutture artificiali permanenti e miglioramenti immobiliari.
LOCATION	Luoghi fisici come strade, città, stati, paesi, specchi d'acqua o coordinate geografiche.
VALORE_MONETARIO	Il valore di qualcosa negli Stati Uniti o in un'altra valuta. Il valore può essere specifico o approssimativo.
ORGANIZZAZIONE	Società e altri gruppi di persone definiti da una struttura organizzativa consolidata.
PERSONA	I nomi o i soprannomi di individui o personaggi immaginari.
PERSON_TITLE	Qualsiasi titolo che descriva una persona, che di solito è una categoria occupazionale (ad esempio CEO) o onorifica (come Mr.).

Tipo	Description
QUANTITÀ	Un numero o un valore e l'unità di misura.
STOCK_CODE	Un simbolo azionario, ad esempio AMZN, un codice ISIN (International Securities Identification Number), il Committee on Uniform Securities Identification Procedures (CUSIP) o la Stock Exchange Daily Official List (SEDOL).

Event types (Tipi di evento)

Tipo	Description
BANCAROTTA	Un procedimento legale che coinvolge una persona o una società che non è in grado di rimborsare i debiti insoluti.
OCCUPAZIONE	Si verifica quando un dipendente viene assunto, licenziato, in pensione o cambia in altro modo lo stato di impiego.
ACQUISIZIONE_AZIENDALE	Si verifica quando una società ottiene il possesso della maggior parte o della totalità delle azioni o dei beni fisici di un'altra società per acquisirne il controllo.
INVESTIMENTO_GENERALE	Si verifica quando una persona o una società acquista un bene con la prospettiva di generare entrate o apprezzamenti futuri.
FUSIONE_AZIENDALE	Si verifica quando due o più società si uniscono per creare una nuova entità legale.
IPO	Un'offerta pubblica iniziale (IPO) di azioni di una società privata al pubblico in una nuova emissione di azioni.

Tipo	Description
EMISSIONE DI DIRITTI	Un gruppo di diritti offerti agli azionisti esistenti per l'acquisto di azioni aggiuntive, noti come warrant di sottoscrizione, in proporzione alle loro partecipazioni esistenti.
OFFERTA_SECONDARIA	Un'offerta di titoli da parte di un azionista di una società.
OFFERTA_SCAFFALE	Una disposizione della Securities and Exchange Commission (SEC) che consente a un emittente di registrare una nuova emissione di titoli e vendere parti dell'emissione per un periodo di tempo senza registrare nuovamente il titolo o incorrere in sanzioni. Nota anche come registrazione sugli scaffali.
OFFERTA_OFFERTA	Un'offerta per l'acquisto di alcune o tutte le azioni degli azionisti di una società.
STOCK_SPLIT	Si verifica quando il consiglio di amministrazione di una società aumenta il numero di azioni in circolazione emettendo più azioni agli attuali azionisti. Questo evento si applica anche ai frazionamenti azionari inversi.

Tipi di argomenti

Tipi di argomenti a favore della bancarotta

Tipo di argomento	Description
FILER	La persona o la società che dichiara il fallimento.
DATE	La data o l'ora del fallimento.

Tipo di argomento	Description
POSTO	ubicazione o struttura in cui (o più vicina a dove) è avvenuta la bancarotta.

Tipi di argomenti a favore dell'OCCUPAZIONE

Tipo	Description
DIPENDENTE	La persona impiegata da un'azienda.
EMPLOYEE_TITLE	Il titolo del dipendente.
DATORE DI LAVORO	La persona o l'azienda che impiega il dipendente.
START_DATE	La data o l'ora di inizio del rapporto di lavoro.
DATA_FINE	La data o l'ora di fine del rapporto di lavoro.

Tipi di argomenti per CORPORATE_ACQUISITION, INVESTMENT_GENERAL

Tipo	Description
QUANTITÀ	Il valore monetario associato alla transazione.
OGGETTO DI INVESTIMENTO	La persona o la società associata all'investimento.
INVESTITORE	La persona o la società che investe nell'asset.
DATE	La data o l'ora dell'acquisizione o dell'investimento.
POSTO	Luogo in cui (o più vicino a dove) ha avuto luogo l'acquisizione o l'investimento.

Tipi di argomenti per CORPORATE_MERGER

Tipo	Description
DATE	La data o l'ora della fusione.
NUOVA_AZIENDA	La nuova entità legale risultante dalla fusione.
PARTECIPANTE	La società coinvolta nella fusione.

Tipi di argomenti per IPO, RIGHTS_ISSUE, SECONDARY_OFFERING, SHELF_OFFERING, TENDER_OFFERING

Tipo	Description
DATA_SCADENZA	La data o l'ora di scadenza dell'offerta.
INVESTITORE	La persona o la società che investe nell'asset.
EMITTENTE	La persona o la società che riceve l'offerta.
OFFERING_AMOUNT	Il valore monetario associato all'offerta.
DATA_OFFERTA	La data o l'ora dell'offerta.
OFFERENTE	La persona o la società che avvia l'offerta.
VALORE_TOTAL_OFFERENTE	Il valore monetario totale associato all'offerta.
RECORD_DATE	La data o l'ora di registrazione dell'offerta.
SELLING_AGENT	La persona o la società che facilita la vendita dell'offerta.
PREZZO_AZIONE	Il valore monetario associato al prezzo delle azioni.
CONDIVIDI_QUANTITÀ	Il numero di azioni associate all'offerta.
SOTTOSCRITTORI	La società associata alla sottoscrizione dell'offerta.

Tipi di argomenti per STOCK_SPLIT

Tipo	Description
COMPAGNIA	La società emittente le azioni si fraziona.
DATE	La data o l'ora del frazionamento azionario.
SPLIT_RATIO	Il rapporto tra l'aumento del numero di nuove azioni in circolazione e il numero attuale di azioni prima del frazionamento azionario.

Frase chiave

Una frase chiave è una stringa contenente una frase sostantiva che descrive una cosa particolare. In genere è costituita da un sostantivo e dai modificatori che lo contraddistinguono. Ad esempio, «giorno» è un sostantivo; «una bella giornata» è un sostantivo che include un articolo («a») e un aggettivo («bello»). Ogni frase chiave include un punteggio che indica il livello di fiducia di Amazon Comprehend rispetto al fatto che la stringa sia una frase sostantiva. Puoi utilizzare il punteggio per determinare se il rilevamento ha un livello di confidenza sufficientemente elevato per la tua applicazione.

Le operazioni di rilevamento delle frasi chiave possono essere eseguite utilizzando una qualsiasi delle lingue principali supportate da Amazon Comprehend. Tutti i documenti devono essere nella stessa lingua.

È possibile utilizzare una delle seguenti operazioni per rilevare le frasi chiave in un documento o in un set di documenti.

- [DetectKeyPhrases](#)
- [BatchDetectKeyPhrases](#)
- [StartKeyPhrasesDetectionJob](#)

Le operazioni restituiscono un elenco di [KeyPhrase](#) oggetti, uno per ogni frase chiave del documento. L'`BatchDetectKeyPhrases` operazione restituisce un elenco di `KeyPhrase` oggetti, uno per ogni documento del batch. L'`StartKeyPhrasesDetectionJob` operazione avvia un processo asincrono che produce un file contenente un elenco di `KeyPhrase` oggetti per ogni documento del lavoro.

L'esempio seguente è la risposta dell'operazione. DetectKeyPhrases

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

Lingua dominante

Puoi usare Amazon Comprehend per esaminare il testo e determinare la lingua dominante. Amazon Comprehend identifica la lingua utilizzando identificatori di RFC 5646: se esiste un identificatore ISO 639-1 a 2 lettere, con un sottotag regionale, se necessario, lo utilizza. Altrimenti, utilizza il codice ISO 639-2 a 3 lettere.

Per ulteriori informazioni su RFC 5646, vedere [Tag per l'identificazione delle lingue](#) sul sito Web IETF Tools.

La risposta include un punteggio che indica il livello di confidenza di Amazon Comprehend rispetto al fatto che una particolare lingua è la lingua dominante nel documento. Ogni punteggio è indipendente dagli altri punteggi. Il punteggio non indica che una lingua costituisca una percentuale particolare di un documento.

Se un documento lungo (come un libro) contiene più lingue, puoi suddividere il documento lungo in parti più piccole ed eseguire l'DetectDominantLanguageoperazione sulle singole parti. È quindi possibile aggregare i risultati per determinare la percentuale di ciascuna lingua nel documento più lungo.

Il rilevamento del linguaggio Amazon Comprehend presenta le seguenti limitazioni:

- Non supporta il rilevamento fonetico del linguaggio. Ad esempio, non rileva «arigato» come giapponese o «nihao» come cinese.
- Può avere difficoltà a distinguere coppie linguistiche simili, come l'indonesiano e il malese, o il bosniaco, il croato e il serbo.
- Per ottenere risultati ottimali, fornite almeno 20 caratteri di testo di input.

Amazon Comprehend rileva le seguenti lingue.

Codice	Lingua
af	Afrikaans
am	Amarico
ar	Arabo
as	Assamese
az	Azero
ba	Bashkir
be	Bielorusso
bn	Bengalese
bs	Bosniaco
bg	Bulgaro
ca	catalano
ceb	Cebuano
cs	Ceco
cv	Chuvash
cy	Gallese

Codice	Lingua
da	Danese
de	Tedesco
el	Greco
en	Italiano
eo	Esperanto
et	Estone
eu	Basco
fa	Persiano
fi	Finlandese
fr	Francese
gd	Gaelico scozzese
ga	Irlandese
gl	Galiziano
gu	Gujarati
ht	Haitiano
he	Ebraico
ha	Hausa
hi	Hindi
hr	Croato
hu	Ungherese

Codice	Lingua
hy	Armeno
ilo	Iloko
id	Indonesiano
is	Islandese
it	Italiano
jv	Javanese
ja	Giapponese
kn	Kannada
ka	Georgiano
kk	Kazako
km	Khmer centrale
ky	Kirghiso
ko	Coreano
ku	Curda
lo	Lao
la	Latino
lv	Lettone
lt	Lituano
lb	Lussemburghese
ml	Malese

Codice	Lingua
mt	Maltese
mr	Marathi
mk	Macedone
mg	Malgascio
mn	Mongolo
ms	Malese
my	Birmano
ne	Nepalese
new	Newari
nl	Olandese
no	Norvegese
or	Oriya
om	Oromo
pa	Punjabi
pl	Polacco
pt	Portoghese
ps	Pushto
qu	Quechua
ro	Romeno
ru	Russo

Codice	Lingua
sa	Sanscrito
si	Singalese
sk	Slovacco
sl	Sloveno
sd	Sindhi
so	Somalo
es	Spagnolo
sq	Albanese
sr	Serbo
su	Sundanese
sw	Swahili
sv	Svedese
ta	Tamil
tt	Tartaro
te	Telugu
tg	Tagiko
tl	Tagalog
th	Thai
tk	Turkmeno
tr	Turco

Codice	Lingua
ug	Uiguro
uk	Ucraino
ur	Urdu
uz	Uzbeko
vi	Vietnamita
yi	Yiddish
yo	Yoruba
zh	Cinese (semplificato)
zh-TW	Cinese (tradizionale)

È possibile utilizzare una delle seguenti operazioni per rilevare la lingua dominante in un documento o in un set di documenti.

- [DetectDominantLanguage](#)
- [BatchDetectDominantLanguage](#)
- [StartDominantLanguageDetectionJob](#)

L'`DetectDominantLanguage` operazione restituisce un [DominantLanguage](#) oggetto.

L'`BatchDetectDominantLanguage` operazione restituisce un elenco di `DominantLanguage` oggetti, uno per ogni documento del batch. L'`StartDominantLanguageDetectionJob` operazione avvia un lavoro asincrono che produce un file contenente un elenco di `DominantLanguage` oggetti, uno per ogni documento del lavoro.

L'esempio seguente è la risposta dell'operazione. `DetectDominantLanguage`

```
{
  "Languages": [
    {
      "LanguageCode": "en",
```

```
    "Score": 0.9793661236763
  }
]
```

Sentiment

Usa Amazon Comprehend per determinare il sentimento dei contenuti nei documenti di testo con codifica UTF-8. Ad esempio, puoi utilizzare l'analisi del sentiment per determinare i sentimenti dei commenti su un post di blog e determinare se ai tuoi lettori è piaciuto il post.

Puoi determinare il sentiment dei documenti in una qualsiasi delle lingue principali supportate da Amazon Comprehend. Tutti i documenti di un lavoro devono essere nella stessa lingua.

La determinazione del sentiment restituisce i seguenti valori:

- Positivo: il testo esprime un sentimento complessivamente positivo.
- Negativo: il testo esprime un sentimento complessivamente negativo.
- Misto: il testo esprime sentimenti sia positivi che negativi.
- Neutro: il testo non esprime sentimenti né positivi né negativi.

Puoi utilizzare una qualsiasi delle seguenti operazioni API per rilevare il sentimento di un documento o di un insieme di documenti.

- [DetectSentiment](#)
- [BatchDetectSentiment](#)
- [StartSentimentDetectionJob](#)

Le operazioni restituiscono il sentimento più probabile per il testo e i punteggi per ciascuno dei sentimenti. Il punteggio rappresenta la probabilità che l'emozione sia stata rilevata correttamente. Ad esempio, nell'esempio seguente è probabile al 95% che il testo abbia un sentimento. *Positive* C'è meno dell'1% di probabilità che il testo abbia un *Negative* sentimento. È possibile utilizzare il *SentimentScore* per determinare se la precisione del rilevamento soddisfa le esigenze dell'applicazione.

L'*DetectSentiment* operazione restituisce un oggetto che contiene il sentimento rilevato e un [SentimentScore](#) oggetto. L'*BatchDetectSentiment* operazione restituisce un elenco di sentimenti e *SentimentScore* oggetti, uno per ogni documento del batch.

L'operazione `StartSentimentDetectionJob` avvia un processo asincrono che produce un file contenente un elenco di sentimenti e `SentimentScore` oggetti, uno per ogni documento del lavoro.

L'esempio seguente è la risposta dell'operazione. `DetectSentiment`

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

Sentimento mirato

Il sentimento mirato fornisce una comprensione dettagliata dei sentimenti associati a entità specifiche (come marchi o prodotti) nei documenti di input.

La differenza tra il sentimento mirato e il [sentimento](#) è il livello di granularità dei dati di output. L'analisi del sentiment determina il sentiment dominante per ogni documento di input, ma non fornisce dati per ulteriori analisi. L'analisi mirata del sentiment determina il sentimento a livello di entità per entità specifiche in ogni documento di input. È possibile analizzare i dati di output per determinare i prodotti e i servizi specifici che ottengono feedback positivi o negativi.

Ad esempio, in una serie di recensioni di ristoranti, un cliente fornisce la seguente recensione: «I tacos erano deliziosi e il personale era cordiale». L'analisi di questa recensione produce i seguenti risultati:

- L'analisi del sentiment determina se il sentimento complessivo di ogni recensione di un ristorante è positivo, negativo, neutro o misto. In questo esempio, il sentimento generale è positivo.
- L'analisi mirata del sentiment determina il sentiment per le entità e gli attributi del ristorante menzionati dai clienti nelle recensioni. In questo esempio, il cliente ha espresso commenti positivi su «tacos» e «staff».

Targeted Sentiment fornisce i seguenti risultati per ogni lavoro di analisi:

- Identità delle entità menzionate nei documenti.

- Classificazione del tipo di entità per ogni entità menzionata.
- Il sentimento e un punteggio di sentimento per ciascuna menzione dell'entità.
- Gruppi di menzioni (gruppi di co-riferimento) che corrispondono a una singola entità.

Puoi utilizzare la [console](#) o l'[API](#) per eseguire analisi mirate del sentiment. La console e l'API supportano l'analisi in tempo reale e l'analisi asincrona per un sentiment mirato.

Amazon Comprehend supporta il sentiment mirato per i documenti in lingua inglese.

Per ulteriori informazioni sul sentiment mirato, incluso un tutorial, consulta [Extract granular sentiment in text with Amazon Comprehend Targeted Sentiment nel blog](#) sull'apprendimento automatico. AWS

Argomenti

- [Tipi di entità](#)
- [Gruppo di co-riferimento](#)
- [Organizzazione dei file di output](#)
- [Analisi in tempo reale tramite console](#)
- [Esempio di produzione mirata del sentiment](#)

Tipi di entità

Il sentiment mirato identifica i seguenti tipi di entità. Assegna il tipo di entità OTHER se l'entità non appartiene a nessun'altra categoria. Ogni entità menzionata nel file di output include il tipo di entità, ad esempio "Type": "PERSON".

Definizioni dei tipi di entità

Tipo di entità	Definizione
PERSONA	Gli esempi includono individui, gruppi di persone, soprannomi, personaggi immaginari e nomi di animali.
LOCATION	Luoghi geografici come paesi, città, stati, indirizzi, formazioni geologiche, specchi d'acqua, monumenti naturali e luoghi astronomici.
ORGANIZZAZIONE	Gli esempi includono governi, aziende, squadre sportive e religioni.

Tipo di entità	Definizione
SERVIZIO, STRUTTURA	Edifici, aeroporti, autostrade, ponti e altre strutture artificiali permanenti e miglioramenti immobiliari.
MARCA	Organizzazione, gruppo o produttore di uno specifico articolo o linea di prodotti commerciale.
OGGETTO_COMMERCIALE	Qualsiasi articolo acquistabile o acquisibile non generico, compresi i veicoli e i prodotti di grandi dimensioni per i quali è stato prodotto un solo articolo.
FILM	Un film o un programma televisivo. Entity potrebbe essere il nome completo, un soprannome o un sottotitolo.
MUSICA	Una canzone, completa o parziale. Inoltre, raccolte di singole creazioni musicali, come un album o un'antologia.
LIBRO	Un libro, pubblicato professionalmente o autopubblicato.
SOFTWARE	Un prodotto software rilasciato ufficialmente.
GIOCO	Un gioco, ad esempio videogiochi, giochi da tavolo, giochi comuni o sport.
TITOLO_PERSONALE	Titoli ufficiali e onorificenze come Presidente, PhD o Dr.
EVENT	Gli esempi includono festival, concerti, elezioni, guerre, conferenze ed eventi promozionali.
DATE	Qualsiasi riferimento a una data o a un'ora, specifica o generale, assoluta o relativa.
QUANTITÀ	Tutte le misurazioni con le relative unità (valuta, percentuale, numero, byte, ecc.).
ATTRIBUTE	Un attributo, una caratteristica o un tratto di un'entità, ad esempio la «qualità» di un prodotto, il «prezzo» di un telefono o la «velocità» di una CPU.
OTHER	Entità che non appartengono a nessuna delle altre categorie.

Gruppo di co-riferimento

Targeted Sentiment identifica i gruppi di co-riferimento in ogni documento di input. Un gruppo di co-riferimento è un gruppo di menzioni in un documento che corrispondono a un'entità del mondo reale.

Example

Nel seguente esempio di recensione di un cliente, «spa» è l'entità con il tipo di entità. FACILITY. L'entità ha due menzioni aggiuntive come pronomi («it»).



Organizzazione dei file di output

Il lavoro mirato di analisi del sentiment crea un file di output di testo JSON. Il file contiene un oggetto JSON per ciascuno dei documenti di input. Ogni oggetto JSON contiene i seguenti campi:

- Entità: una serie di entità trovate nel documento.
- File: il nome del file del documento di input.
- Riga: se il file di input è un documento per riga, Entities contiene il numero di riga del documento nel file.

Note

Se Targeted Sentiment non identifica alcuna entità nel testo di input, restituisce un array vuoto come risultato Entities.

L'esempio seguente mostra Entities per un file di input con tre righe di input. Il formato di input è ONE_DOC_PER_LINE, quindi ogni riga di input è un 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
}
```

Un'entità nell'array Entities include un raggruppamento logico (chiamato gruppo di co-riferimento) delle menzioni di entità rilevate nel documento. Ogni entità ha la seguente struttura generale:

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

Un'entità contiene i seguenti campi:

- **Menzioni:** una serie di menzioni dell'entità nel documento. L'array rappresenta un gruppo di co-riferimento. Consulta [the section called “Gruppo di co-riferimento”](#) per un esempio. L'ordine delle menzioni nell'array Mentions è l'ordine della loro posizione (offset) nel documento. Ogni menzione include il punteggio del sentiment e il punteggio di gruppo per quella menzione. Il punteggio di gruppo indica il livello di fiducia che queste menzioni appartengano alla stessa entità.

- **DescriptiveMentionIndex**— Uno o più indici nell'array **Mentions** che forniscono il nome migliore per il gruppo di entità. Ad esempio, un'entità potrebbe avere tre menzioni con valori di testo «ABC Hotel», «ABC Hotel» e «it». Il nome migliore è «ABC Hotel», che ha un **DescriptiveMentionIndex** valore di [0,1].

Ogni menzione include i seguenti campi

- **BeginOffset**— L'offset nel testo del documento in cui inizia la menzione.
- **EndOffset**— L'offset rispetto al testo del documento in cui termina la menzione.
- **GroupScore**— La fiducia che tutte le entità menzionate nel gruppo si riferiscano alla stessa entità.
- **Testo**: il testo del documento che identifica l'entità.
- **Tipo**: il tipo di entità. Amazon Comprehend supporta diversi tipi di [entità](#).
- **Punteggio**: modello di fiducia che l'entità sia pertinente. L'intervallo di valori va da zero a uno, dove uno rappresenta la massima confidenza.
- **MentionSentiment**— Contiene il sentiment e il punteggio di sentiment per la menzione.
- **Sentimento**: il sentimento della menzione. I valori includono: POSITIVO, NEUTRO, NEGATIVO e MISTO.
- **SentimentScore**— Fornisce un modello di fiducia per ciascuno dei sentimenti possibili. L'intervallo di valori è compreso tra zero e uno corrisponde alla massima confidenza.

I valori **Sentiment** hanno il seguente significato:

- **Positivo**: l'entità menzionata esprime un sentimento positivo.
- **Negativo**: la menzione dell'entità esprime un sentimento negativo.
- **Mista**: la menzione dell'entità esprime sentimenti sia positivi che negativi.
- **Neutro**: la menzione dell'entità non esprime sentimenti positivi o negativi.

Nell'esempio seguente, un'entità ha una sola menzione nel documento di input, quindi **DescriptiveMentionIndex** è zero (la prima menzione nell'array **Mentions**). L'entità identificata è una **PERSONA** con il nome «I». Il punteggio del sentiment è neutro.

```
{"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
}
```

Analisi in tempo reale tramite console

Puoi utilizzare la console Amazon Comprehend per l'esecuzione [the section called “Sentimento mirato”](#) in tempo reale. Usa il testo di esempio o incolla il tuo testo nella casella di testo di input, quindi scegli Analizza.

Nel pannello Insights, la console mostra tre visualizzazioni dell'analisi mirata del sentiment:

- **Testo analizzato:** visualizza il testo analizzato e sottolinea ogni entità. Il colore della sottolineatura indica il valore di sentiment (positivo, neutro, negativo o misto) assegnato dall'analisi all'entità. La console visualizza le mappature dei colori nell'angolo superiore destro della casella di testo analizzata. Se si posiziona il cursore su un'entità, la console visualizza un pannello popup contenente i valori di analisi (tipo di entità, punteggio di sentiment) relativi all'entità.
- **Risultati:** visualizza una tabella contenente una riga per ogni menzione di entità identificata nel testo. Per ogni entità, la tabella mostra l'[entità](#) e il punteggio dell'entità. La riga include anche il sentimento principale e il punteggio per ogni valore del sentiment. Se sono presenti più menzioni

della stessa entità, nota come [the section called “Gruppo di co-riferimento”](#), la tabella visualizza queste menzioni come un insieme di righe comprimibile associato all'entità principale.

Se passi il mouse su una riga di entità nella tabella Risultati, la console evidenzia l'entità menzionata nel pannello di testo Analizzato.

- Integrazione delle applicazioni: visualizza i valori dei parametri della richiesta API e la struttura dell'oggetto JSON restituito nella risposta API. Per una descrizione dei campi dell'oggetto JSON, consulta [the section called “Organizzazione dei file di output”](#)

Esempio di analisi in tempo reale da console

Questo esempio utilizza il testo seguente come input, che è il testo di input predefinito fornito dalla 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.
```

Il pannello Testo analizzato mostra il seguente output per questo esempio. Passa il mouse sul testo Zhang Wei per visualizzare il pannello popup relativo a questa entità.

Insights [Info](#)

Entities

Key phrases

Language

PII

Sentiment

Targeted sentiment

Syntax

Analyzed text

Positive

Neutral

Negative

Mixed

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 by July 31st. Based on your autopay settings, we will withdraw your payment on the due date from your bank account XXXXXX1111 with the routing number XXXXX0000. Sunshine Spa, 123 Main St, Anywhere. Send comments to Alice at sunspa@mail.com. The spa was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great place to visit.

Entity type: PERSON

Entity confidence: 0.99+

Sentiment: NEUTRAL

Sentiment confidence: 0.99+

Total related entities: 5

La tabella Risultati fornisce dettagli aggiuntivi su ciascuna entità, tra cui il punteggio dell'entità, il sentimento principale e il punteggio per ogni sentimento.

▼ Results

Q Search < 1 2 >						
Entity	Entity type	Entity score	Primary sentiment	Positive score	Negative score	
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	-	-	

Nel nostro esempio, l'analisi mirata del sentiment riconosce che ogni tua menzione nel testo di input è un riferimento all'entità personale Zhang Wei. La console visualizza queste menzioni come un insieme di righe comprimibili associate all'entità principale.

▼ Results

Search < 1 2 > ⚙️						
Entity	Entity type	Entity score	Primary sentiment	Positive score	Ne	
☒ Zhang Wei (5)	PERSON	-	— NEUTRAL	-	-	
☐ your	PERSON	0.99+	— NEUTRAL	0	0	
☐ your	PERSON	0.67	— NEUTRAL	0	0	
☐ Your	ORGANIZATION	0.94	— NEUTRAL	0	0	
☐ your	PERSON	0.99+	— NEUTRAL	0	0	
☐ Zhang Wei	PERSON	0.99+	— NEUTRAL	0.00	0	

Il pannello di integrazione dell'applicazione mostra l'oggetto JSON generato dall' DetectTargetedSentiment API. Per un esempio completo, consultate la sezione seguente.

Esempio di produzione mirata del sentiment

L'esempio seguente mostra il file di output di un lavoro mirato di analisi del sentiment. Il file di input è composto da tre semplici documenti:

The burger was very flavorful and the burger bun was excellent. However, customer service was slow.

My burger was good, and it was warm. The burger had plenty of toppings.

The burger was cooked perfectly but it was cold. The service was OK.

L'analisi mirata del sentiment di questo file di input produce il seguente output.

```
{
  "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": 0.000008,
            "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
}
}

```

Analisi della sintassi

Usa l'analisi della sintassi per analizzare le parole del documento e restituire la parte del discorso, o funzione sintattica, per ogni parola del documento. È possibile identificare i nomi, i verbi, gli aggettivi e così via nel documento. Utilizzate queste informazioni per ottenere una comprensione più approfondita del contenuto dei documenti e per comprendere la relazione tra le parole nel documento.

Ad esempio, è possibile cercare i sostantivi in un documento e quindi cercare i verbi correlati a tali sostantivi. In una frase come «Mia nonna ha spostato il suo divano» puoi vedere i sostantivi «nonna» e «divano» e il verbo «si è trasferita». Puoi utilizzare queste informazioni per creare applicazioni per l'analisi del testo per le combinazioni di parole che ti interessano.

Per iniziare l'analisi, Amazon Comprehend analizza il testo di origine per trovare le singole parole nel testo. Dopo l'analisi del testo, a ogni parola viene assegnata la parte del discorso che riprende nel testo sorgente.

Amazon Comprehend è in grado di identificare le seguenti parti del discorso.

Token	Parte del discorso
AGG	Aggettivo Parole che in genere modificano i sostantivi.
ADP	Adozione La testa di una frase preposizionale o postposizionale.
AVV	Avverbio Parole che in genere modificano i verbi. Possono anche modificare aggettivi e altri avverbi.
AUX	Ausiliario Parole funzionali che accompagnano il verbo di una frase verbale.
CCONJ	Congiunzione coordinativa Una congiunzione di coordinamento collega parole,

Token	Parte del discorso
	frasi o clausole di una frase senza subordinare l'una all'altra.
CONJ	Conjunction Una congiunzione collega parole, frasi o clausole in una frase.
DIETA	Determinante Articoli e altre parole che specificano una particolare frase nominale.
INTJ	Interiezione Parole usate come esclamazione o parte di un'esclamazione.
SOSTANTIVO	Sostantivo Parole che specificano una persona, un luogo, una cosa, un animale o un'idea.
NUM	Numerale Parole, in genere determinanti, aggettivi o pronomi, che esprimono un numero.

Token	Parte del discorso
O	Altro Parole a cui non è possibile assegnare una parte della categoria vocale.
PART	Particle Funzione di parole associate a un'altra parola o frase per conferire significato.
GREMBIULE	Pronome Parole che sostituiscono i sostantivi o le frasi nominali.
OGGETTO DI SCENA	Nome proprio Un nome che è il nome di un individuo, luogo o oggetto specifico.
PUNTO	punteggiatura Caratteri non alfabetici che delimitano il testo.
SCONJ	Congiunzione subordinante Una congiunzione che unisce una clausola dipendente a una frase. Un esempio di congiunzione subordinata è «perché».

Token	Parte del discorso
SIMMETRIA	Symbol Entità simili a parole come il simbolo del dollaro (\$) o simboli matematici.
VERBO	Verb Parole che segnalano eventi e azioni.

Per ulteriori informazioni sulle parti del discorso, consulta i [tag Universal POS](#) sul sito Web Universal Dependencies.

Le operazioni restituiscono token che identificano la parola e la parte del discorso che la parola rappresenta nel testo. Ogni token rappresenta una parola nel testo sorgente. Fornisce la posizione della parola nella fonte, la parte del discorso che la parola assume nel testo, la sicurezza di Amazon Comprehend che la parte del discorso sia stata identificata correttamente e la parola che è stata analizzata dal testo di origine.

Di seguito è riportata la struttura dell'elenco dei token di sintassi. Viene generato un token di sintassi per ogni parola del documento.

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

Ogni token fornisce le seguenti informazioni:

- **BeginOffset** e **EndOffset**: fornisce la posizione della parola nel testo di input.
- **PartOfSpeech**—Fornisce due informazioni, una **Tag** che identifica la parte del discorso e l'altra **Score** che rappresenta la fiducia che Amazon Comprehend Syntax ha nel fatto che la parte del discorso sia stata identificata correttamente.
- **Text**—Fornisce la parola che è stata identificata.
- **TokenId**—Fornisce un identificatore per il token. L'identificatore è la posizione del token nell'elenco dei token.

Amazon Comprehend personalizzato

Puoi personalizzare Amazon Comprehend in base ai tuoi requisiti specifici senza le competenze necessarie per creare soluzioni NLP basate sull'apprendimento automatico. Utilizzando l'apprendimento automatico, o AutoML, Comprehend Custom crea modelli NLP personalizzati per conto dell'utente, utilizzando i dati di formazione forniti dall'utente.

Elaborazione dei documenti di input: Amazon Comprehend supporta l'elaborazione dei documenti in un'unica fase per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Ad esempio, è possibile inserire una combinazione di documenti di testo semplice e documenti semistrutturati (come documenti PDF, documenti Microsoft Word e immagini) in un lavoro di analisi personalizzato. Per ulteriori informazioni, consulta [Elaborazione dei documenti](#).

Classificazione personalizzata: crea modelli di classificazione personalizzati (classificatori) per organizzare i documenti in categorie personalizzate. Per ogni etichetta di classificazione, fornisci una serie di documenti che la rappresentino al meglio e addestra il tuo classificatore a utilizzarla. Una volta addestrato, un classificatore può essere utilizzato su un numero qualsiasi di set di documenti senza etichetta. Puoi utilizzare la console per un'esperienza senza codice o installare l'SDK più recente. AWS Per ulteriori informazioni, consulta [Classificazione personalizzata](#).

Riconoscimento personalizzato delle entità: crea modelli di riconoscimento delle entità personalizzati (riconoscitori) in grado di analizzare il testo in base a termini specifici e frasi basate su sostantivi. Puoi addestrare gli addetti al riconoscimento a estrarre termini come numeri di polizze o frasi che implicano un aumento del numero di clienti. Per addestrare il modello, è necessario fornire un elenco delle entità e una serie di documenti che le contengono. Una volta addestrato il modello, è possibile inviare lavori di analisi in base al modello per estrarne le entità personalizzate. Per ulteriori informazioni, consulta [Riconoscimento personalizzato delle entità](#).

Modellazione degli argomenti

Puoi usare Amazon Comprehend per esaminare il contenuto di una raccolta di documenti per determinare temi comuni. Ad esempio, puoi fornire ad Amazon Comprehend una raccolta di articoli di notizie che determinerà gli argomenti, come sport, politica o intrattenimento. Il testo nei documenti non deve essere annotato.

Amazon Comprehend utilizza un modello di apprendimento basato sull'[allocazione di dirichlet latenti](#) per determinare gli argomenti in un set di documenti. Esamina ogni documento per determinare il contesto e il significato di una parola. L'insieme di parole che spesso appartengono allo stesso contesto nell'intero set di documenti costituisce un argomento.

Una parola viene associata a un argomento in un documento in base alla prevalenza di tale argomento in un documento e all'affinità dell'argomento con la parola. La stessa parola può essere associata a diversi argomenti in diversi documenti in base alla distribuzione degli argomenti in un determinato documento.

Ad esempio, la parola «glucosio» in un articolo che parla principalmente di sport può essere assegnata all'argomento «sport», mentre la stessa parola in un articolo sulla «medicina» verrà assegnata all'argomento «medicina».

A ogni parola associata a un argomento viene assegnato un peso che indica in che misura la parola contribuisce a definire l'argomento. Il peso è un'indicazione di quante volte la parola ricorre nell'argomento rispetto alle altre parole dell'argomento, nell'intero set di documenti.

Per ottenere risultati più accurati, dovresti fornire ad Amazon Comprehend il corpus più ampio possibile con cui lavorare. Per ottenere risultati ottimali:

- È necessario utilizzare almeno 1.000 documenti in ogni lavoro di modellazione tematica.
- Ogni documento deve contenere almeno 3 frasi.
- Se un documento è composto principalmente da dati numerici, è necessario rimuoverlo dal corpus.

La modellazione degli argomenti è un processo asincrono. Invii il tuo elenco di documenti ad Amazon Comprehend da un bucket Amazon S3 utilizzando l'operazione. [StartTopicsDetectionJob](#) La risposta viene inviata a un bucket Amazon S3. Puoi configurare sia i bucket di input che di output. Visualizzate un elenco degli argomenti relativi ai lavori di modellazione inviati utilizzando l'[ListTopicsDetectionJobs](#) operazione e visualizzate le informazioni su un lavoro che utilizza l'[DescribeTopicsDetectionJob](#) operazione. I contenuti distribuiti ai bucket Amazon S3 potrebbero

contenere informazioni dei clienti. Per ulteriori informazioni sulla rimozione di dati sensibili, vedi [Come svuotare un bucket S3?](#) o [Come eliminare un bucket S3?](#).

I documenti devono essere in file di testo in formato UTF-8. Puoi inviare i tuoi documenti in due modi. La tabella seguente mostra le opzioni.

Formato	Description
Un documento per file	Ogni file contiene un documento di input. Questa opzione è ideale per raccolte di documenti di grandi dimensioni.
Un documento per riga	L'input è un singolo file. Ogni riga del file è considerata un documento. Questa opzione è ideale per documenti brevi, come i post sui social media. Ogni riga deve terminare con un'alimentazione di linea (LF,\n), un valore di ritorno (CR,\r) o entrambi (CRLF,\r\n). Il separatore di riga Unicode (u+2028) non può essere usato per terminare una riga.

Per ulteriori informazioni, consulta il tipo di dati. [InputDataConfig](#)

Dopo aver elaborato la tua raccolta di documenti, Amazon Comprehend restituisce un archivio compresso contenente due file e. `topic-terms.csv` `doc-topics.csv` Per ulteriori informazioni sul file di output, consulta. [OutputDataConfig](#)

Il primo file di output `topic-terms.csv`, è un elenco di argomenti della raccolta. Per ogni argomento, per impostazione predefinita l'elenco include i termini principali per argomento in base al loro peso. Ad esempio, se offri ad Amazon Comprehend una raccolta di articoli di giornale, potrebbe restituire quanto segue per descrivere i primi due argomenti della raccolta:

Topic	Termine	Weight
000	team	0,118533

Topic	Termine	Weight
000	game	0,106072
000	player (giocatore)	0,031625
000	stagione	0.023633
000	giocare	0,021118
000	cortile	0,024454
000	allenatore	0.016012
000	giochi	0.016191
000	calcio	0.015049
000	terzino	0,014239
001	tazza	0,205236
001	cibo	0.040686
001	minuti	0,036062
001	aggiungi	0,029697
001	cucchiaino	0,028789
001	petrolio	0,021254
001	pepe	0.022205
001	cucchiaino	0,020040
001	vino	0.01658
001	zucchero	0,015101

I pesi rappresentano una distribuzione di probabilità sulle parole di un determinato argomento. Poiché Amazon Comprehend restituisce solo le prime 10 parole per ogni argomento, la somma dei pesi non sarà 1,0. Nei rari casi in cui un argomento contiene meno di 10 parole, la somma dei pesi sarà 1,0.

Le parole vengono ordinate in base al loro potere discriminante osservando la loro presenza in tutti gli argomenti. In genere questo valore corrisponde al loro peso, ma in alcuni casi, ad esempio quando compaiono le parole «play» e «yard» nella tabella, il risultato è un ordine diverso dal peso.

È possibile specificare il numero di argomenti da restituire. Ad esempio, se chiedi ad Amazon Comprehend di restituire 25 argomenti, restituirà i 25 argomenti più importanti della raccolta. Amazon Comprehend è in grado di rilevare fino a 100 argomenti in una raccolta. Scegli il numero di argomenti in base alla tua conoscenza del dominio. Potrebbero essere necessari alcuni esperimenti per arrivare al numero corretto.

Il secondo file (`doc-topics.csv`) elenca i documenti associati a un argomento e la proporzione del documento interessato dall'argomento. Se è stato specificato `ONE_DOC_PER_FILE`, il documento viene identificato dal nome del file. Se è stato specificato, `ONE_DOC_PER_LINE` il documento viene identificato dal nome del file e dal numero di riga indicizzato a 0 all'interno del file. Ad esempio, Amazon Comprehend potrebbe restituire quanto segue per una raccolta di documenti inviati con un documento per file:

Documento	Topic	Proporzione
sample-doc1	000	0,999330137
esempio-doc2	000	0,998532187
esempio-doc3	000	0,998384574
...		
Esempio: DOCN	000	3,57E-04

Amazon Comprehend utilizza le informazioni del set di dati delle liste di lemmatizzazione di MBM, reso disponibile [qui](#) con la licenza [Open](#) database (L) v1.0. ODb

Modalità di elaborazione dei documenti

Amazon Comprehend supporta tre modalità di elaborazione dei documenti. La modalità scelta dipende dal numero di documenti da elaborare e dall'immediatezza con cui è necessario visualizzare i risultati:

- Documento singolo sincrono: chiami Amazon Comprehend con un singolo documento e ricevi una risposta sincrona, consegnata immediatamente alla tua applicazione (o alla console).
- Sincrona multi-documento: richiami l'API Amazon Comprehend con una raccolta di un massimo di 25 documenti e ricevi una risposta sincrona.
- Batch asincrono: per un'ampia raccolta di documenti, inserisci i documenti in un bucket Amazon S3 e avvia un processo asincrono (utilizzando operazioni di console o API) per analizzare i documenti. Amazon Comprehend archivia i risultati dell'analisi nell'S3 bucket/folder specificato nella richiesta.

Argomenti

- [Elaborazione di un singolo documento](#)
- [Elaborazione sincrona di più documenti](#)
- [Elaborazione asincrona in batch](#)

Elaborazione di un singolo documento

Le operazioni su un singolo documento sono operazioni sincrone che restituiscono i risultati dell'analisi del documento direttamente all'applicazione. Utilizzate le operazioni sincrone a documento singolo quando create un'applicazione interattiva che funziona su un documento alla volta.

Per ulteriori informazioni sulle operazioni sincrone dell'API, vedere [Analisi in tempo reale utilizzando i modelli integrati](#) (per console) e [Analisi in tempo reale tramite l'API](#)

Elaborazione sincrona di più documenti

Se desideri elaborare più documenti, puoi utilizzare le operazioni Batch* API per inviare più di un documento ad Amazon Comprehend alla volta. Puoi inviare fino a 25 documenti per richiesta. Amazon Comprehend restituisce un elenco di risposte, una per ogni documento della richiesta. Le richieste effettuate con queste operazioni sono sincrone. L'applicazione richiama l'operazione e quindi attende la risposta dal servizio.

L'utilizzo Batch* delle operazioni è identico alla chiamata del singolo documento APIs per ciascuno dei documenti della richiesta. Il loro utilizzo APIs può comportare prestazioni migliori per le applicazioni.

L'input di ciascuno di essi APIs è una struttura JSON contenente i documenti da elaborare. Per tutte le operazioni tranne `BatchDetectDominantLanguage`, è necessario impostare la lingua di input. È possibile impostare una sola lingua di input per ogni richiesta. Ad esempio, quanto segue è l'input dell'`BatchDetectEntities` operazione. Contiene due documenti ed è in inglese.

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

La risposta di un'Batch*operazione contiene due elenchi, il `ResultList` e il `ErrorList`. `ResultList` contiene un record per ogni documento che è stato elaborato correttamente. Il risultato per ogni documento della richiesta è identico al risultato che si otterrebbe eseguendo una singola operazione sul documento. Ai risultati di ogni documento viene assegnato un indice basato sull'ordine dei documenti nel file di input. La risposta dell'`BatchDetectEntities` operazione è:

```
{
  "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 si verifica un errore nella richiesta, la risposta contiene un messaggio `ErrorList` che identifica i documenti che contenevano l'errore. Il documento è identificato dal relativo indice nell'elenco di input. Ad esempio, il seguente input dell'`BatchDetectLanguage` operazione contiene un documento che non può essere elaborato:

```

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

```

La risposta di Amazon Comprehend include un elenco di errori che identifica il documento che conteneva un errore:

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

Per ulteriori informazioni sulle operazioni sincrone dell'API in batch, consulta [Batch in tempo reale APIs](#)

Elaborazione asincrona in batch

Per analizzare documenti di grandi dimensioni e grandi raccolte di documenti, utilizza le operazioni asincrone di Amazon Comprehend.

Per analizzare una raccolta di documenti, in genere esegui i seguenti passaggi:

1. Archivia i documenti in un bucket Amazon S3.
2. Avvia uno o più processi di analisi per analizzare i documenti.

3. Monitora lo stato di avanzamento dei lavori di analisi.
4. Recupera i risultati dell'analisi da un bucket S3 quando il lavoro è completo.

Per ulteriori informazioni sull'utilizzo delle operazioni API asincrone, consulta (console) e [Esecuzione di processi di analisi utilizzando la console](#) [Lavori di analisi asincrona utilizzando l'API](#)

Lingue supportate in Amazon Comprehend

Amazon Comprehend supporta un'ampia varietà di lingue per le sue varie funzionalità. Le lingue supportate e le funzionalità che le supportano sono riportate nelle tabelle seguenti.

Argomenti

- [Lingue supportate](#)
- [Lingue supportate dalle funzionalità di Amazon Comprehend](#)

Lingue supportate

Amazon Comprehend (ad eccezione della funzionalità di rilevamento della lingua dominante) supporta le seguenti lingue per una o più funzionalità.

Codice	Lingua
de	Tedesco
en	Italiano
es	Spagnolo
it	Italiano
pt	Portoghese
fr	Francese
ja	Giapponese
ko	Coreano
hi	Hindi
ar	Arabo
zh	Cinese (semplificato)
zh-TW	Cinese (tradizionale)

Note

Amazon Comprehend identifica la lingua utilizzando identificatori di RFC 5646, se esiste un identificatore ISO 639-1 a 2 lettere, con un sottotag regionale./ Se necessario, lo utilizza.

Altrimenti, utilizza il codice ISO 639-2 a 3 lettere.

Per ulteriori informazioni su RFC 5646, vedere [Tag per l'identificazione delle lingue](#) sul sito Web IETF Tools.

Lingue supportate dalle funzionalità di Amazon Comprehend

Funzionalità	Lingue supportate
Lingua dominante	Per informazioni, consulta Lingua dominante .
Entità	Tutte le lingue supportate.
Frase chiave	Tutte le lingue supportate.
Rilevamento delle entità PII	Inglese e spagnolo.
Etichettatura delle entità PII	Inglese e spagnolo.
Sentiment	Tutte le lingue supportate.
Sentimento mirato	Inglese.
Analisi della sintassi	tedesco (de), inglese (en), spagnolo (es), francese (fr), italiano (it) e portoghese (pt).
Modellazione degli argomenti	Non dipende dalla lingua utilizzata. Non supporta lingue basate su caratteri come cinese, giapponese e coreano.
Classificazione personalizzata	I modelli in testo semplice supportano le seguenti lingue: tedesco (de), inglese (en), spagnolo (es), francese (fr), italiano (it) e portoghese (pt).

Funzionalità	Lingue supportate
	<u>I modelli di documenti nativi supportano solo documenti</u> in inglese.
<u>Riconoscimento personalizzato delle entità</u>	Tedesco (de), inglese (en), spagnolo (es), francese (fr), italiano (it) e portoghese (pt). Custom Entity Recognition per PDF e Word supporta solo documenti in inglese.

Configurazione

Prima di utilizzare Amazon Comprehend per la prima volta, completa le seguenti attività.

Configurazione delle attività

- [Iscriviti per un Account AWS](#)
- [Crea un utente con accesso amministrativo](#)
- [Configura il AWS Command Line Interface \(AWS CLI\)](#)
- [Concessione dell'accesso programmatico](#)

Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la <https://portal.aws.amazon.com/billing/registrazione>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com/> e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [Console di gestione AWS](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Accedere come utente root](#) nella Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita il Centro identità IAM.

Per istruzioni, consulta [Abilitazione del AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Nel Centro identità IAM, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

Assegnazione dell'accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

Configura il AWS Command Line Interface (AWS CLI)

Non è necessario AWS CLI eseguire i passaggi degli esercizi introduttivi. Tuttavia, è richiesto per altri esercizi presenti in questa guida. Se preferisci, puoi saltare questo passaggio e procedere alla [Guida introduttiva ad Amazon Comprehend](#) configurazione AWS CLI successiva.

Per installare e configurare il AWS CLI

1. Installa il AWS CLI. Per istruzioni, consulta il seguente argomento nella Guida AWS Command Line Interface per l'utente:

[Installazione o aggiornamento della versione più recente di AWS Command Line Interface](#)

2. Configurare AWS CLI. Per istruzioni, consultate il seguente argomento nella Guida AWS Command Line Interface per l'utente:

[Configurazione della AWS Command Line Interface](#)

Concessione dell'accesso programmatico

Gli utenti necessitano di un accesso programmatico se desiderano interagire con l' AWS esterno di. Console di gestione AWS Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come	Segui le istruzioni per l'interfaccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
	credenziali temporanee per firmare le richieste programmatiche a,, o. AWS CLI AWS SDKs AWS APIs	• Per la AWS CLI, consulta Login for AWS local development nella Guida per l'AWS Command Line Interface utente. • Per AWS SDKs, consulta Login for AWS local development nella AWS SDKs and Tools Reference Guide.
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporanee per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, vedere Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente.AWS Command Line Interface • Per AWS SDKs gli strumenti e AWS APIs, consulta l'autenticazione di IAM Identity Center nella Guida di riferimento AWS SDKs and Tools.
IAM	Utilizza credenziali temporanee per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'utente IAM .

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli strumenti AWS SDKs e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti. • Per AWS APIs, consulta la sezione Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Guida introduttiva ad Amazon Comprehend

L'esercizio seguente utilizza la console Amazon Comprehend per creare ed eseguire un processo di rilevamento di entità asincrono. Questo esercizio presuppone che tu abbia dimestichezza con Amazon Simple Storage Service (Amazon S3). Per un esempio più semplice, vedi. [Analisi in tempo reale utilizzando i modelli integrati](#)

Per creare un processo di rilevamento delle entità

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analysis Jobs, quindi scegli Crea lavoro.
3. In Impostazioni Job, assegna un nome al lavoro. Il nome deve essere univoco all'interno della regione e dell'account.
4. Per Tipo di analisi, scegli Entità.
5. Per Lingua, scegliete la lingua dei documenti di input.
6. In Dati di input, per Origine dati, scegli Documenti di esempio. La console imposta la posizione S3 come cartella contenente gli esempi pubblici.
7. In Dati di output, nella posizione S3, incolla l'URL o la posizione della cartella in Amazon S3 per i file di output.
8. Nella sezione Autorizzazioni di accesso, seleziona Crea un ruolo IAM. La console crea un nuovo ruolo IAM con le autorizzazioni appropriate per Amazon Comprehend per accedere ai bucket di input e output.
9. Una volta completata la compilazione del modulo, scegli Crea lavoro per creare e avviare il processo di rilevamento degli argomenti.

Il nuovo lavoro viene visualizzato nell'elenco dei lavori con il campo di stato che mostra lo stato del lavoro. Il campo può essere IN_PROGRESS relativo a un lavoro in fase COMPLETED di elaborazione, a un lavoro terminato correttamente e FAILED a un lavoro con un errore.

10. Scegli il lavoro per aprire il pannello Dettagli del lavoro.
11. In Output, in Output data location scegli il link per aprire la console Amazon S3.
12. Nella console Amazon S3, scegli Scarica e salva il output.tar.gz file.
13. Decomprimi il file e salvalo come file Json.

14. Vedi [the section called “Entità”](#) per una descrizione dei tipi di entità e dei campi per ogni entità rilevata.

Analisi tramite la console Amazon Comprehend

Puoi utilizzare la console Amazon Comprehend per analizzare documenti in tempo reale o eseguire processi di analisi asincroni.

Utilizzando l'analisi in tempo reale con modelli integrati, puoi riconoscere entità, estrarre frasi chiave, rilevare la lingua principale, rilevare informazioni personali, determinare il sentimento, analizzare il sentimento mirato e analizzare la sintassi.

Puoi eseguire lavori di analisi utilizzando i modelli integrati per trovare informazioni approfondite come entità, eventi, frasi, lingua principale, sentimento, sentimento mirato e informazioni di identificazione personale (PII). Puoi anche eseguire lavori di modellazione di argomenti.

La console supporta anche l'analisi in tempo reale e asincrona utilizzando modelli personalizzati. Per ulteriori informazioni, consultare [Classificazione personalizzata](#) e [Riconoscimento personalizzato delle entità](#).

Argomenti

- [Analisi in tempo reale utilizzando i modelli integrati](#)
- [Esecuzione di processi di analisi utilizzando la console](#)

Analisi in tempo reale utilizzando i modelli integrati

Puoi utilizzare la console Amazon Comprehend per eseguire analisi in tempo reale di un documento di testo con codifica UTF-8. Il documento può essere in inglese o in una delle altre lingue supportate da Amazon Comprehend. I risultati vengono visualizzati nella console in modo da poter rivedere l'analisi.

Per iniziare ad analizzare i documenti, accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).

Puoi sostituire il testo di esempio con il tuo testo e quindi scegliere Analizza per eseguire un'analisi del testo. Sotto il testo da analizzare, il riquadro Risultati mostra ulteriori informazioni sul testo.

Esegui analisi in tempo reale utilizzando il modello integrato

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>

2. Dal menu a sinistra, scegli Analisi in tempo reale.
3. In Tipo di input, scegli Integrato per il tipo di analisi.
4. Inserisci il testo che desideri analizzare.
5. Scegli Analizza. La console visualizza i risultati dell'analisi del testo nel pannello Insights. Il pannello Insights include una scheda per ciascuno dei tipi di approfondimenti. Le seguenti sezioni descrivono i risultati per il tipo di approfondimento.

Argomenti

- [Entità](#)
- [Frase chiave](#)
- [Lingua](#)
- [Informazioni personali di identificazione \(PII\)](#)
- [Sentiment](#)
- [Sentimento mirato](#)
- [Sintassi](#)

Entità

La scheda Entità elenca ogni entità, la relativa categoria e il livello di confidenza rilevato da Amazon Comprehend nel testo di input. I risultati sono codificati a colori per indicare diversi tipi di entità come organizzazioni, sedi, date e persone. Per ulteriori informazioni, consulta [Entità](#).

Insights **Info**

Entities | Key phrases | Language | PII | Sentiment | Targeted sentiment | Syntax

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

< 1 2 > ⚙

Entity	Type	Confidence
Zhang Wei	Person	0.99+
John	Person	0.99+
AnyCompany Financial Services, LLC	Organization	0.99+
1111-0000-1111-0008	Other	0.99+
\$24.53	Quantity	0.99+
July 31st	Date	0.99+
XXXXXX1111	Other	0.98
XXXXX0000	Other	0.96
Sunshine Spa	Organization	0.98
123 Main St	Location	0.98

► Application integration

Frasi chiave

La scheda Frasi chiave elenca le frasi nominali chiave rilevate da Amazon Comprehend nel testo di input e il livello di confidenza associato. Per ulteriori informazioni, consulta [Frasi chiave](#).

Insights [Info](#)

Entities

Key phrases

Language

PII

Sentiment

Targeted sentiment

Syntax

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

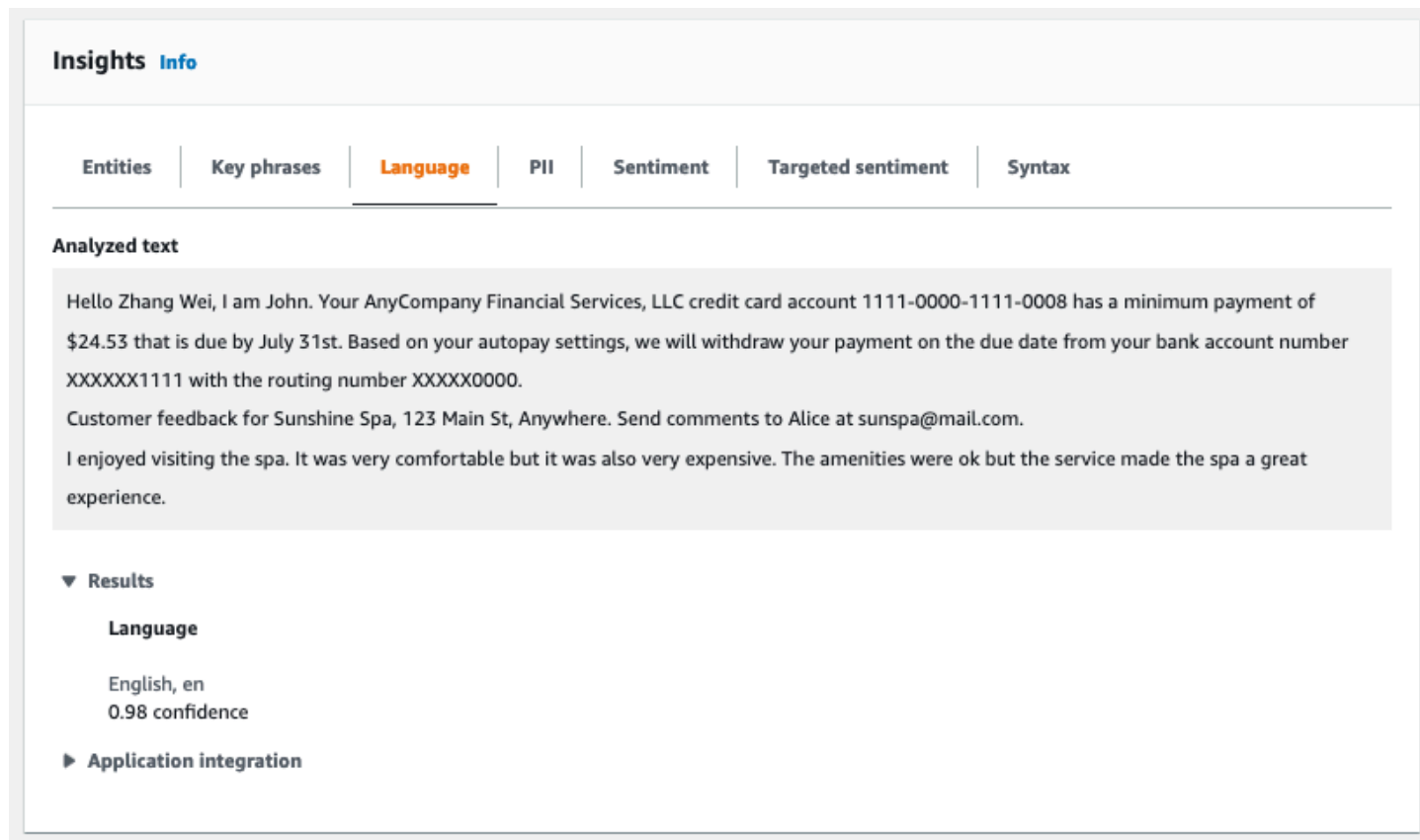
< 1 2 3 >

Key phrases	Confidence
Zhang Wei	0.93
John	0.99+
Your AnyCompany Financial Services	0.98
LLC credit card account 1111-0000-1111-0008	0.87
a minimum payment	0.99+
\$24.53	0.99+
July 31st	0.99+
your autopay settings	0.99+
your payment	0.99+
the due date	0.99+

► Application integration

Lingua

La scheda Lingua mostra la lingua dominante del testo e il livello di fiducia di Amazon Comprehend nel rilevare correttamente la lingua dominante. Amazon Comprehend è in grado di riconoscere 100 lingue. Per ulteriori informazioni, consulta [Lingua dominante](#).



The screenshot displays the Amazon Comprehend Insights interface. At the top, there's a header with 'Insights' and an 'Info' link. Below this is a navigation bar with tabs: 'Entities', 'Key phrases', 'Language' (which is selected and highlighted in orange), 'PII', 'Sentiment', 'Targeted sentiment', and 'Syntax'. The main content area is titled 'Analyzed text' and contains three paragraphs of sample text. Below the text, there's a 'Results' section with a dropdown arrow. Under 'Results', the 'Language' section is expanded, showing 'English, en' with a '0.98 confidence' score. At the bottom of the results section, there's a link for 'Application integration'.

Insights [Info](#)

Entities | Key phrases | **Language** | PII | Sentiment | Targeted sentiment | Syntax

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**

Language

English, en
0.98 confidence

► [Application integration](#)

Informazioni personali di identificazione (PII)

La scheda PII elenca le entità nel testo di input che contengono informazioni di identificazione personale (PII). Un'entità PII è un riferimento testuale ai dati personali che potrebbero essere utilizzati per identificare un individuo, come un indirizzo, un numero di conto bancario o un numero di telefono. Per ulteriori informazioni, consulta [Rilevamento delle entità PII](#).

La scheda PII offre due modalità di analisi:

- Offset
- Etichette

Offset

La modalità di analisi Offsets identifica la posizione delle PII nei documenti di testo. Per ulteriori informazioni, consulta [Individua le entità PII](#).

InsightsInfo

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

Q 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

Etichette

La modalità di analisi Labels verifica la presenza di PII nel documento di testo e restituisce le etichette dei tipi di entità PII identificati. Per ulteriori informazioni, consulta [Etichettatura delle entità PII](#).

The screenshot displays the Amazon Comprehend Insights interface. At the top, there's a header with 'Insights' and an 'Info' link. Below this is a navigation bar with tabs: 'Entities', 'Key phrases', 'Language', 'PII' (which is selected and highlighted in orange), 'Sentiment', 'Targeted sentiment', and 'Syntax'. The main content area is titled 'Personally identifiable information (PII) analysis mode'. It contains two radio buttons: 'Offsets' (unselected) and 'Labels' (selected). Below the radio buttons, there's a section titled 'Results' with a search bar and a table. The table has two columns: 'Type' and 'Confidence'. The table lists five types of PII with their corresponding confidence scores: Email (1.00), Name (0.99+), Bank account number (0.98), Bank routing (0.69), and Credit debit number (1.00). At the bottom of the interface, there's a link for 'Application integration'.

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.

▼ Results

Search

Type	Confidence
Email	1.00
Name	0.99+
Bank account number	0.98
Bank routing	0.69
Credit debit number	1.00

► Application integration

Sentiment

La scheda Sentiment mostra il sentimento dominante del testo. Il sentimento può essere valutato come neutro, positivo, negativo o misto. In questo caso, ogni sentiment ha un indice di fiducia, che fornisce una stima di Amazon Comprehend secondo cui tale sentiment è dominante. Per ulteriori informazioni, consulta [Sentiment](#).

The screenshot displays the Amazon Comprehend Insights console. At the top, there's a header with 'Insights' and an 'Info' link. Below this is a navigation bar with tabs: 'Entities', 'Key phrases', 'Language', 'PII', 'Sentiment' (which is selected and highlighted in orange), 'Targeted sentiment', and 'Syntax'. The main content area is titled 'Analyzed text' and contains three paragraphs of sample text. Below the text, there's a section labeled 'Results' with a dropdown arrow. Under 'Results', the 'Sentiment' section is expanded, showing four categories: 'Neutral' (0.56 confidence), 'Positive' (0.10 confidence), 'Negative' (0.19 confidence), and 'Mixed' (0.14 confidence). At the bottom of the 'Results' section, there's a link for 'Application integration'.

Insights Info

Entities | Key phrases | Language | PII | **Sentiment** | Targeted sentiment | Syntax

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

Sentiment

Neutral	Positive	Negative	Mixed
0.56 confidence	0.10 confidence	0.19 confidence	0.14 confidence

► Application integration

Sentimento mirato

L'analisi mirata del sentiment identifica i sentimenti espressi sulle entità citate nel testo. Amazon Comprehend assegna un punteggio di sentiment a ogni menzione di un'entità, insieme a un indice di fiducia e ad altre informazioni. Una valutazione del sentiment può essere neutra, positiva, negativa o mista.

Nel pannello Testo analizzato, la console sottolinea ciascuna delle entità analizzate. Il colore del testo sottolineato indica il sentimento generale dell'entità. Se si posiziona il cursore su un'entità, la console visualizza informazioni aggiuntive in una finestra pop-up.

Insights [Info](#)

Entities

Key phrases

Language

PII

Sentiment

Targeted sentiment

Syntax

Analyzed text

Positive

Neutral

Negative

Mixed

Hello Zhang Wei , I am John . Your AnyCompany Financial Services, LLC credit card account 1111-0000-1111-0008 has a minimum payment 31st . Based on your autopay settings, we will withdraw your payment on the due date from your X1111 with the routing number XXXXX0000.

Entity type: PERSON

Entity confidence: 0.99+

Sentiment: NEUTRAL

Sentiment confidence: 0.99+

Total related entities: 5

ine Spa , 123 Main St , Anywhere . Send comments to Alice at sunspa@mail.com .

was very comfortable but it was also very expensive. The amenities were ok but the service made the spa a great

La tabella Risultati fornisce dettagli aggiuntivi su ciascuna entità. Se sono presenti più menzioni della stessa entità, denominate gruppo di co-riferimento, la tabella visualizza queste menzioni come un insieme comprimibile di righe associate all'entità principale.

Nell'esempio seguente, l'entità è una persona di nome Zhang Wei. L'analisi mirata del sentimento riconosce che ogni tua menzione è un riferimento alla stessa persona. La console visualizza queste menzioni come voci secondarie dell'entità principale.

Analyzed text

■ Positive
■ Neutral
■ Negative
■ Mixed

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 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	

Se il testo che stai analizzando non include alcun sentimento mirato [Tipi di entità](#), l'analisi del sentiment mirato visualizza un campo dei risultati vuoto.

Per ulteriori informazioni su come utilizzare la console per l'analisi mirata del sentiment in tempo reale, consulta. [Analisi in tempo reale tramite console](#)

Sintassi

La scheda Sintassi mostra una suddivisione di ogni elemento del testo, insieme alla relativa parte del discorso e al punteggio di confidenza associato. Per ulteriori informazioni, consulta [Analisi della sintassi](#).

Insights [Info](#)

[Entities](#) | [Key phrases](#) | [Language](#) | [PII](#) | [Sentiment](#) | [Targeted sentiment](#) | **[Syntax](#)**

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

< 1 2 3 4 5 6 7 ... 11 >

Word	Part of speech	Confidence
Hello	Interjection	0.98
Zhang	Proper noun	0.99+
Wei	Proper noun	0.99+
,	Punctuation	0.99+
I	Pronoun	0.99+
am	Verb	0.98
John	Proper noun	0.99+
.	Punctuation	0.99+
Your	Pronoun	0.99+
AnyCompany	Proper noun	0.99+

▶ **Application integration**

Esecuzione di processi di analisi utilizzando la console

Puoi utilizzare la console Amazon Comprehend per creare e gestire processi di analisi asincroni. Il tuo lavoro analizza i documenti archiviati in Amazon S3 per trovare entità come eventi, frasi, lingua principale, sentimenti o informazioni di identificazione personale (PII).

Per creare un lavoro di analisi

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analysis jobs, quindi scegli Crea job.
3. In Impostazioni Job, assegna al job di analisi un nome univoco.
4. Per Tipo di analisi, scegliete uno dei tipi di analisi incorporati.

Se scegli Lingua principale o Modellazione tematica, puoi saltare il passaggio successivo.

5. A seconda del tipo di analisi scelto, la console visualizza uno o più dei seguenti campi aggiuntivi:
 - La lingua è obbligatoria per tutti i tipi di analisi integrati ad eccezione della lingua principale e della modellazione degli argomenti.

Scegliete la lingua dei documenti di input.

- I tipi di eventi Target sono obbligatori per il tipo di analisi Eventi.

Seleziona i tipi di eventi da rilevare nei documenti di input. Per ulteriori informazioni sui tipi di eventi supportati, consulta [Event types \(Tipi di evento\)](#).

- Le impostazioni di rilevamento PII sono necessarie per il tipo di analisi PII.

Seleziona la modalità di output. Per ulteriori informazioni sulle impostazioni di rilevamento dei dati PII, vedere [Rilevamento delle entità PII](#).

6. In Dati di input, specifica dove si trovano i documenti di input in Amazon S3:
 - Per analizzare i tuoi documenti, scegli I miei documenti e scegli Sfoglia S3 per fornire il percorso del bucket o della cartella che contiene i tuoi file.
 - Per analizzare i campioni forniti da Amazon Comprehend, scegli Documenti di esempio. In questo caso, Amazon Comprehend utilizza un bucket gestito da AWS e tu non ne specifichi la posizione.
7. (Facoltativo) Per il formato di input, specifica uno dei seguenti formati per i tuoi file di input:
 - Un documento per file: ogni file contiene un documento di input. Questa soluzione è ideale per raccolte di documenti di grandi dimensioni.
 - Un documento per riga: l'input è costituito da uno o più file. Ogni riga di un file è considerata un documento. Questa opzione è ideale per documenti brevi, come i post sui social media. Ogni riga deve terminare con un'alimentazione di linea (LF,\n), un valore di ritorno (CR,\r)

o entrambi (CRLF,\ r\n). Non è possibile utilizzare il separatore di riga UTF-8 (u+2028) per terminare una riga.

8. In Dati di output, scegli Browse S3. Scegli il bucket o la cartella Amazon S3 in cui desideri che Amazon Comprehend scriva i dati di output prodotti dall'analisi.
9. (Facoltativo) Per crittografare il risultato dell'output del tuo lavoro, scegli Crittografia. Quindi, scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account:
 - Se utilizzi una chiave associata all'account corrente, scegli l'alias o l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'alias o l'ID della chiave in KMS key ID.

 Note

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta [Servizio di gestione delle chiavi](#) (KMS).

10. In Autorizzazioni di accesso, fornisci un ruolo IAM che:
 - Garantisce l'accesso in lettura alla posizione Amazon S3 dei documenti di input.
 - Concede l'accesso in scrittura alla posizione Amazon S3 dei documenti di output.
 - Include una politica di fiducia che consente al responsabile del `comprehend.amazonaws.com` servizio di assumere il ruolo e ottenere le relative autorizzazioni.

Se non disponi già di un ruolo IAM con queste autorizzazioni e una policy di fiducia appropriata, scegli Crea un ruolo IAM per crearne uno.

11. Quando hai finito di compilare il modulo, scegli Crea lavoro per creare e avviare il processo di rilevamento degli argomenti.

Il nuovo lavoro viene visualizzato nell'elenco dei lavori con il campo di stato che mostra lo stato del lavoro. Il campo può essere IN_PROGRESS relativo a un lavoro in fase COMPLETED di elaborazione, a un lavoro terminato correttamente e FAILED a un lavoro con un errore. È possibile fare clic su un lavoro per ottenere ulteriori informazioni sul lavoro, inclusi eventuali messaggi di errore.

Una volta completato il processo, Amazon Comprehend archivia i risultati dell'analisi nella posizione di output Amazon S3 specificata per il lavoro. Per una descrizione dei risultati dell'analisi per ogni tipo di analisi, consulta. [Approfondimenti](#)

Utilizzo dell'API Amazon Comprehend

L'API Amazon Comprehend supporta le operazioni per eseguire analisi in tempo reale (sincrone) e operazioni per avviare e gestire processi di analisi asincroni.

Puoi utilizzare direttamente gli operatori dell'API Amazon Comprehend oppure puoi utilizzare la CLI o una delle SDKs. Gli esempi in questo capitolo utilizzano la CLI, l'SDK Python e l'SDK Java.

Per eseguire gli esempi AWS CLI e Python, è necessario installare AWS CLI. Per ulteriori informazioni, consulta [Configura il AWS Command Line Interface \(AWS CLI\)](#).

Per eseguire gli esempi Java, è necessario installare AWS SDK per Java. Per istruzioni sull'installazione dell'SDK per Java, [consulta Configurare AWS l'SDK per Java](#).

Argomenti

- [Utilizzo di Amazon Comprehend con un SDK AWS](#)
- [Analisi in tempo reale tramite l'API](#)
- [Lavori di analisi asincrona utilizzando l'API](#)

Utilizzo di Amazon Comprehend con un SDK AWS

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
AWS SDK per Rust	AWS SDK per Rust esempi di codice
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Fornisci un feedback nella parte inferiore di questa pagina.

Analisi in tempo reale tramite l'API

Gli esempi seguenti mostrano come utilizzare l'API Amazon Comprehend per l'analisi in tempo reale, utilizzando e AWS SDKs per .NET, Java e Python. AWS CLI Usa gli esempi per conoscere le operazioni sincrone di Amazon Comprehend e come elementi costitutivi per le tue applicazioni.

Negli esempi di .NET riportati in questa sezione viene utilizzato [AWS SDK per .NET](#). Puoi usarli [AWS Toolkit for Visual Studio](#) per sviluppare AWS applicazioni usando .NET. Include modelli utili e AWS Explorer per la distribuzione di applicazioni e la gestione dei servizi. Per una prospettiva da sviluppatore .NET AWS, consulta la [AWS guida per gli sviluppatori .NET](#).

Argomenti

- [Rilevare la lingua dominante](#)
- [Rilevamento di entità denominate](#)
- [Rilevamento delle frasi chiave](#)
- [Determinare il sentimento](#)
- [Analisi in tempo reale per un sentiment mirato](#)
- [Rilevamento della sintassi](#)
- [Batch in tempo reale APIs](#)

Rilevare la lingua dominante

Per determinare la lingua dominante utilizzata nel testo, utilizzare l'[DetectDominantLanguage](#) operazione. Per rilevare la lingua dominante in un massimo di 25 documenti in un batch, utilizzate l'[BatchDetectDominantLanguage](#) operazione. Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#).

Argomenti

- [Usando il AWS Command Line Interface](#)
- [Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET](#)

Usando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'[DetectDominantLanguage](#) operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

Amazon Comprehend risponde con quanto segue:

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

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

Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET

Per esempi SDK su come determinare la lingua dominante, consulta. [Utilizzo DetectDominantLanguage con un AWS SDK o una CLI](#)

Rilevamento di entità denominate

Per determinare le entità denominate in un documento, utilizzate l'[DetectEntities](#) operazione. Per rilevare entità in un massimo di 25 documenti in un batch, utilizzate l'[BatchDetectEntities](#) operazione. Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#).

Argomenti

- [Usando il AWS Command Line Interface](#)
- [Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET](#)

Usando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'[DetectEntities](#) operazione utilizzando. AWS CLI È necessario specificare la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

Amazon Comprehend risponde con quanto segue:

```
{  
  "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"
}
```

Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET

Per esempi SDK su come determinare la lingua dominante, consulta. [Utilizzo DetectEntities con un AWS SDK o una CLI](#)

Rilevamento delle frasi chiave

Per determinare i sostantivi chiave usati nel testo, usate l'operazione. [DetectKeyPhrases](#) Per rilevare i sostantivi chiave in un massimo di 25 documenti in un batch, utilizzate l'operazione. [BatchDetectKeyPhrases](#) Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#).

Argomenti

- [Usando il AWS Command Line Interface](#)
- [Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET](#)

Usando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'`DetectKeyPhrases` operazione con. AWS CLI È necessario specificare la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

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

Amazon Comprehend risponde con quanto segue:

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

Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET

Per esempi SDK che rilevano frasi chiave, consulta. [Utilizzo DetectKeyPhrases con un AWS SDK o una CLI](#)

Determinare il sentimento

Amazon Comprehend fornisce le seguenti operazioni API per l'analisi del sentiment:

- [DetectSentiment](#)— Determina il sentimento emotivo generale di un documento.
- [BatchDetectSentiment](#)— Determina il sentimento generale in un massimo di 25 documenti in batch. Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#)
- [StartSentimentDetectionJob](#)— Avvia un processo asincrono di rilevamento del sentiment per una raccolta di documenti.
- [ListSentimentDetectionJobs](#)— Restituisce l'elenco dei lavori di rilevamento dei sentimenti che hai inviato.

- [DescribeSentimentDetectionJob](#)— Ottiene le proprietà (incluso lo stato) associate al lavoro di rilevamento del sentiment specificato.
- [StopSentimentDetectionJob](#)— Interrompe il processo di sentiment in corso specificato.

Argomenti

- [Usando il AWS Command Line Interface](#)
- [Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET](#)

Usando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'`DetectSentiment` operazione con AWS CLI. Questo esempio specifica la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

Amazon Comprehend risponde con quanto segue:

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

Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET

Per esempi SDK che determinano il sentimento del testo in ingresso, consulta [Utilizzo DetectSentiment con un AWS SDK o una CLI](#)

Analisi in tempo reale per un sentiment mirato

Amazon Comprehend fornisce le seguenti operazioni API per l'analisi mirata del sentiment in tempo reale:

- [DetectTargetedSentiment](#)— Analizza il sentiment delle entità menzionate in un documento.
- [BatchDetectTargetedSentiment](#)— Analizza il sentiment mirato per un massimo di 25 documenti in batch. Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#)

Se il testo che stai analizzando non include alcun sentimento mirato [Tipi di entità](#), l'API restituisce un array Entities vuoto.

Usando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'`DetectTargetedSentiment` operazione con AWS CLI. Questo esempio specifica la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

Amazon Comprehend risponde con quanto segue:

```
{
  "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
    }
}
]
}
]
}

```

Rilevamento della sintassi

Per analizzare il testo per estrarre le singole parole e determinare le parti del discorso per ogni parola, usa l'operazione. [DetectSyntax](#) Per analizzare la sintassi di un massimo di 25 documenti in un batch, utilizzate l'operazione. [BatchDetectSyntax](#) Per ulteriori informazioni, consulta [Batch in tempo reale APIs](#).

Argomenti

- [Usando il. AWS Command Line Interface](#)
- [Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET](#)

Usando il. AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'DetectSyntaxoperazione con. AWS CLI Questo esempio specifica la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```

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

```

Amazon Comprehend risponde con quanto segue:

```

{
  "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
    }
]
}

```

Utilizzando l' AWS SDK per Java SDK per Python o SDK per .NET

Per esempi SDK che rilevano la sintassi del testo di input, consulta. [Utilizzo DetectSyntax con un AWS SDK o una CLI](#)

Batch in tempo reale APIs

Per inviare batch composti da un massimo di 25 documenti, puoi utilizzare le operazioni batch in tempo reale di Amazon Comprehend. La chiamata di un'operazione batch è identica alla chiamata del singolo documento APIs per ogni documento della richiesta. L'utilizzo del batch APIs può migliorare le prestazioni delle applicazioni. Per ulteriori informazioni, consulta [Elaborazione sincrona di più documenti](#).

Argomenti

- [Elaborazione in batch con AWS CLI](#)
- [Elaborazione in batch con AWS SDK per .NET](#)

Elaborazione in batch con AWS CLI

Questi esempi mostrano come utilizzare le operazioni dell'API batch utilizzando AWS Command Line Interface. Tutte le operazioni tranne BatchDetectDominantLanguage l'utilizzo del seguente file JSON process.json chiamato input. Per tale operazione l'LanguageCode entità non è inclusa.

Il terzo documento nel file JSON ("\$\$\$\$\$\$\$\$") causerà un errore durante l'elaborazione in batch. È incluso in modo che le operazioni includano un [BatchItemError](#) nella risposta.

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

Gli esempi sono formattati per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Argomenti

- [Rileva la lingua dominante usando un batch \(\)AWS CLI](#)
- [Rileva le entità utilizzando un batch \(AWS CLI\)](#)
- [Rileva le frasi chiave usando un batch \(\)AWS CLI](#)
- [Rileva il sentimento usando un batch \(\)AWS CLI](#)

Rileva la lingua dominante usando un batch ()AWS CLI

L'[BatchDetectDominantLanguage](#) operazione determina la lingua dominante di ogni documento in un batch. Per un elenco delle lingue che Amazon Comprehend è in grado di rilevare, consulta. [Lingua dominante](#) Il AWS CLI comando seguente richiama l'BatchDetectDominantLanguage operazione.

```
aws comprehend batch-detect-dominant-language \
```

```
--endpoint endpoint \  
--region region \  
--cli-input-json file://path to input file/process.json
```

Di seguito è riportata la risposta dell'BatchDetectDominantLanguageoperazione:

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

Rileva le entità utilizzando un batch (AWS CLI)

Utilizzate l'[BatchDetectEntities](#)operazione per trovare le entità presenti in un batch di documenti. Per ulteriori informazioni sulle entità, consulta [Entità](#). Il AWS CLI comando seguente richiama l'BatchDetectEntitiesoperazione.

```
aws comprehend batch-detect-entities \  

```

```
--endpoint endpoint \  
--region region \  
--cli-input-json file://path to input file/process.json
```

Rileva le frasi chiave usando un batch ()AWS CLI

L'[BatchDetectKeyPhrases](#) operazione restituisce le frasi nominali chiave in un batch di documenti. Il AWS CLI comando seguente richiama l'operazione. BatchDetectKeyNounPhrases

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

Rileva il sentimento usando un batch ()AWS CLI

Rileva il sentimento generale di un batch di documenti utilizzando l'[BatchDetectSentiment](#) operazione. Il AWS CLI comando seguente richiama l'BatchDetectSentiment operazione.

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

Elaborazione in batch con AWS SDK per .NET

Il seguente programma di esempio mostra come utilizzare l'[BatchDetectEntities](#) operazione con SDK per .NET. La risposta del server contiene un [BatchDetectEntitiesItemResult](#) oggetto per ogni documento che è stato elaborato correttamente. Se si verifica un errore nell'elaborazione di un documento, nella risposta verrà visualizzato un record nell'elenco degli errori. L'esempio recupera ogni documento con un errore e li invia nuovamente.

L'esempio.NET in questa sezione utilizza il [AWS SDK per .NET](#). È possibile utilizzarli [AWS Toolkit for Visual Studio](#) per sviluppare AWS applicazioni utilizzando .NET. Include modelli utili e AWS Explorer per la distribuzione di applicazioni e la gestione dei servizi. Per una prospettiva da sviluppatore.NET AWS, consulta la [AWS guida per gli sviluppatori.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.USSouth2);

            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");
}
}
}
}

```

Lavori di analisi asincrona utilizzando l'API

I seguenti esempi utilizzano Amazon Comprehend asynchronous APIs per creare e gestire processi di analisi, utilizzando AWS CLI

Argomenti

- [Analisi asincrona per Amazon Comprehend Insights](#)
- [Analisi asincrona per un sentiment mirato](#)

- [Analisi asincrona per il rilevamento degli eventi](#)
- [Analisi asincrona per la modellazione degli argomenti](#)

Analisi asincrona per Amazon Comprehend Insights

Le seguenti sezioni utilizzano l'API Amazon Comprehend per eseguire operazioni asincrone per analizzare le informazioni di Amazon Comprehend.

Argomenti

- [Prerequisiti](#)
- [Avvio di un lavoro di analisi](#)
- [Monitoraggio dei lavori di analisi](#)
- [Ottenere i risultati dell'analisi](#)

Prerequisiti

I documenti devono essere in file di testo in formato UTF-8. Puoi inviare i tuoi documenti in due formati. Il formato utilizzato dipende dal tipo di documenti che si desidera analizzare, come descritto nella tabella seguente.

Description	Formato
Ogni file contiene un documento di input. Questa opzione è ideale per raccolte di documenti di grandi dimensioni.	Un documento per file
L'input è costituito da uno o più file. Ogni riga di un file è considerata un documento. Questa opzione è ideale per documenti brevi, come i post sui social media. Ogni riga deve terminare con un'alimentazione di linea (LF,\n), un valore di ritorno (CR,\r) o entrambi (CRLF,\r\n). Non è possibile utilizzare il separatore di riga UTF-8 (u+2028) per terminare una riga.	Un documento per riga

Quando si avvia un lavoro di analisi, si specifica la posizione S3 per i dati di input. L'URI deve trovarsi nella stessa AWS regione dell'endpoint API che state chiamando. L'URI può puntare a un singolo file o può essere il prefisso per una raccolta di file di dati. Per ulteriori informazioni, consulta il tipo di [InputDataConfig](#) dati.

Devi concedere ad Amazon Comprehend l'accesso al bucket Amazon S3 che contiene la raccolta di documenti e i file di output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Avvio di un lavoro di analisi

Per inviare un lavoro di analisi, utilizza la console Amazon Comprehend o l'operazione appropriata `Start*`:

- [StartDominantLanguageDetectionJob](#)— Inizia un lavoro per rilevare la lingua dominante in ogni documento della raccolta. Per ulteriori informazioni sulla lingua dominante in un documento, vedere [Lingua dominante](#).
- [StartEntitiesDetectionJob](#)— Avvia un processo per rilevare le entità in ogni documento della raccolta. Per ulteriori informazioni sulle entità, consulta [Entità](#).
- [StartKeyPhrasesDetectionJob](#)— Avvia un lavoro per rilevare le frasi chiave in ogni documento della raccolta. Per ulteriori informazioni sulle frasi chiave, vedere [Frase chiave](#).
- [StartPiiEntitiesDetectionJob](#)— Avvia un processo per rilevare le informazioni di identificazione personale (PII) in ogni documento della raccolta. Per ulteriori informazioni sulle PII, vedere [Rilevamento delle entità PII](#).
- [StartSentimentDetectionJob](#)— Inizia un lavoro per rilevare il sentimento in ogni documento della raccolta. Per ulteriori informazioni sui sentimenti, consulta [Sentiment](#).

Monitoraggio dei lavori di analisi

L'`Start*` operazione restituisce un ID che è possibile utilizzare per monitorare l'avanzamento del lavoro.

Per monitorare l'avanzamento utilizzando l'API, si utilizza una delle due operazioni, a seconda che si desideri monitorare l'avanzamento di un singolo lavoro o di più lavori.

Per monitorare l'avanzamento di un singolo processo di analisi, utilizzate le `Describe*` operazioni. Fornisci l'ID del lavoro restituito dall'`Start*` operazione. La risposta dell'`Describe*` operazione contiene il `JobStatus` campo con lo stato del lavoro.

Per monitorare lo stato di avanzamento di più processi di analisi, utilizzate le `List*` operazioni. `List*` le operazioni restituiscono un elenco di offerte di lavoro che hai inviato ad Amazon Comprehend. La risposta include un `JobStatus` campo per ogni lavoro che indica lo stato del lavoro.

Se il campo dello stato è impostato su `COMPLETED` o `FAILED`, l'elaborazione del lavoro è stata completata.

Per conoscere lo stato dei singoli lavori, utilizzate l'`Describe*` operazione per l'analisi che state eseguendo.

- [DescribeDominantLanguageDetectionJob](#)
- [DescribeEntitiesDetectionJob](#)
- [DescribeKeyPhrasesDetectionJob](#)
- [DescribePiiEntitiesDetectionJob](#)
- [DescribeSentimentDetectionJob](#)

Per ottenere lo stato di più lavori, utilizzate l'`List*` operazione per l'analisi che state eseguendo.

- [ListDominantLanguageDetectionJobs](#)
- [ListEntitiesDetectionJobs](#)
- [ListKeyPhrasesDetectionJobs](#)
- [ListPiiEntitiesDetectionJobs](#)
- [ListSentimentDetectionJobs](#)

Per limitare i risultati ai lavori che soddisfano determinati criteri, utilizzate il parametro `List*` delle operazioni `Filter`. È possibile filtrare in base al nome del lavoro, allo stato del lavoro e alla data e all'ora in cui il lavoro è stato inviato. Per ulteriori informazioni, consulta il `Filter` parametro per ciascuna `List*` operazione nel riferimento all'API Amazon Comprehend.

Ottenere i risultati dell'analisi

Al termine di un processo di analisi, utilizzate un'`Describe*` operazione per ottenere la posizione dei risultati. Se lo stato del lavoro è `COMPLETED`, la risposta include un `OutputDataConfig` campo che contiene un campo con la posizione Amazon S3 del file di output. Il `fileoutput.tar.gz`, è un archivio compresso che contiene i risultati dell'analisi.

Se lo stato di un processo è `FAILED`, la risposta contiene un `Message` campo che descrive il motivo per cui il processo di analisi non è stato completato correttamente.

Per ottenere lo stato dei singoli lavori, utilizzate l'`Describe*` operazione appropriata:

- [DescribeDominantLanguageDetectionJob](#)
- [DescribeEntitiesDetectionJob](#)
- [DescribeKeyPhrasesDetectionJob](#)
- [DescribeSentimentDetectionJob](#)

I risultati vengono restituiti in un unico file, con una struttura JSON per ogni documento. Ogni file di risposta include anche messaggi di errore per qualsiasi lavoro con il campo di stato impostato `FAILED` su.

Ciascuna delle seguenti sezioni mostra esempi di output per i due formati di input.

Ottenere risultati di rilevamento della lingua dominante

Di seguito è riportato un esempio di file di output di un'analisi che ha rilevato la lingua dominante. Il formato dell'input è un documento per riga. Per maggiori informazioni, vedi l'operazione [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}
```

Di seguito è riportato un esempio di output di un'analisi in cui il formato dell'input è un documento per file:

```
{"File": "small_doc", "Languages": [{"LanguageCode": "en", "Score": 0.9728053212165833}, {"LanguageCode": "de", "Score": 0.007670710328966379}, {"LanguageCode": "es", "Score": 0.0028472368139773607}]}
{"File": "huge_doc", "Languages": [{"LanguageCode": "en", "Score": 0.984955906867981}, {"LanguageCode": "de", "Score": 0.0026436643674969673}, {"LanguageCode": "fr", "Score": 0.0014206881169229746}]}
```

Ottenere i risultati del rilevamento delle entità

Di seguito è riportato un esempio di file di output di un'analisi che ha rilevato entità nei documenti. Il formato dell'input è un documento per riga. Per maggiori informazioni, vedi l'operazione [DetectEntities](#). L'output contiene due messaggi di errore, uno per un documento troppo lungo e uno per un documento che non è in formato UTF-8.

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

Di seguito è riportato un esempio di output di un'analisi in cui il formato dell'input è un documento per file. L'output contiene due messaggi di errore, uno per un documento troppo lungo e uno per un documento che non è in formato UTF-8.

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

Ottenere i risultati del rilevamento delle frasi chiave

Di seguito è riportato un esempio di file di output di un'analisi che ha rilevato frasi chiave in un documento. Il formato di input è un documento per riga. Per maggiori informazioni, vedi l'operazione [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}
```

Di seguito è riportato un esempio dell'output di un'analisi in cui il formato dell'input è un documento per file.

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

Ottenimento dei risultati del rilevamento di informazioni di identificazione personale (PII)

Di seguito è riportato un esempio di file di output di un processo di analisi che ha rilevato entità PII nei documenti. Il formato dell'input è un documento per riga.

```
{
  "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": 1
},
{
  "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
}
```

Di seguito è riportato un esempio di output di un'analisi in cui il formato dell'input è un documento per file.

```
{
  "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"
}
```

Ottenere risultati di rilevamento del sentiment

Di seguito è riportato un esempio di file di output di un'analisi che ha rilevato il sentimento espresso in un documento. Include un messaggio di errore perché un documento è troppo lungo. Il formato dell'input è un documento per riga. Per maggiori informazioni, vedi l'operazione [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": 0.008045154623687267
  }
}
```

Di seguito è riportato un esempio dell'output di un'analisi in cui il formato dell'input è un documento per file.


```
{"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 exceeds the limit of 5120 bytes."}
```

Analisi asincrona per un sentiment mirato

Per informazioni sull'analisi in tempo reale di Targeted Sentiment, consulta [the section called “Analisi in tempo reale per un sentiment mirato”](#)

Amazon Comprehend fornisce le seguenti operazioni API per avviare e gestire un'analisi asincrona mirata del sentiment:

- [StartTargetedSentimentDetectionJob](#)— Avvia un processo asincrono mirato di rilevamento del sentiment per una raccolta di documenti.
- [ListTargetedSentimentDetectionJobs](#)— Restituisce l'elenco dei lavori mirati di rilevamento dei sentiment che hai inviato.
- [DescribeTargetedSentimentDetectionJob](#)— Ottiene le proprietà (incluso lo stato) associate al lavoro di rilevamento del sentiment mirato specificato.
- [StopTargetedSentimentDetectionJob](#)— Interrompe il lavoro mirato sul sentiment specifico in corso.

Argomenti

- [Prima di iniziare](#)
- [Analisi del sentiment mirato utilizzando AWS CLI](#)

Prima di iniziare

Prima di iniziare, assicurati di avere:

- Bucket di input e output: identifica i bucket Amazon S3 che desideri utilizzare per input e output. I bucket devono trovarsi nella stessa regione dell'API che stai chiamando.
- Ruolo di servizio IAM: è necessario disporre di un ruolo di servizio IAM con l'autorizzazione ad accedere ai bucket di input e output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Analisi del sentiment mirato utilizzando AWS CLI

L'esempio seguente dimostra l'utilizzo dell'`StartTargetedSentimentDetectionJob` operazione con AWS CLI. Questo esempio specifica la lingua del testo di input.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

Per il `cli-input-json` parametro fornite il percorso di un file JSON che contiene i dati della richiesta, come illustrato nell'esempio seguente.

```
{
  "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 la richiesta di avvio del processo ha avuto esito positivo, riceverai la seguente risposta:

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

Analisi asincrona per il rilevamento degli eventi

Argomenti

- [Prima di iniziare](#)
- [Rileva gli eventi utilizzando il AWS CLI](#)

- [Elenca gli eventi utilizzando AWS CLI](#)
- [Descrivi gli eventi utilizzando AWS CLI](#)
- [Ottieni i risultati del rilevamento degli eventi](#)

Per rilevare gli eventi in un set di documenti, utilizzate [StartEventsDetectionJob](#) per avviare un processo asincrono.

Prima di iniziare

Prima di iniziare, assicuratevi di avere:

- Bucket di input e output: identifica i bucket Amazon S3 che desideri utilizzare per input e output. I bucket devono trovarsi nella stessa regione dell'API che stai chiamando.
- Ruolo di servizio IAM: è necessario disporre di un ruolo di servizio IAM con l'autorizzazione ad accedere ai bucket di input e output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Rileva gli eventi utilizzando il AWS CLI

L'esempio seguente dimostra l'utilizzo dell'[StartEventsDetectionJob](#) operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

Per il `cli-input-json` parametro fornite il percorso di un file JSON che contiene i dati della richiesta, come illustrato nell'esempio seguente.

```
{  
  "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 la richiesta di avvio del processo di rilevamento degli eventi ha avuto esito positivo, riceverai la seguente risposta:

```

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

```

Elenca gli eventi utilizzando AWS CLI

Utilizza l'[ListEventsDetectionJobs](#) operazione per visualizzare un elenco dei lavori di rilevamento degli eventi che hai inviato. L'elenco include informazioni sulle posizioni di input e output utilizzate e sullo stato di ciascun processo di rilevamento. L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

In risposta otterrete un codice JSON simile al seguente:

```
{
```

```

"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"
}

```

Descrivi gli eventi utilizzando AWS CLI

È possibile utilizzare l'[DescribeEventsDetectionJob](#) operazione per ottenere lo stato di un lavoro esistente. L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```

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

```

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

In risposta riceverai il seguente codice 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"
    ]
  }
}
```

Ottieni i risultati del rilevamento degli eventi

Di seguito è riportato un esempio di file di output di un processo di analisi che ha rilevato eventi nei documenti. Il formato dell'input è un documento per riga.

```
{
  "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"
        }
      ]
    },
    {
      "Arguments": [
        {
          "EntityIndex": 2,
          "Role": "INVESTEES",
          "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
}
```

Per ulteriori informazioni sulla struttura del file di output degli eventi e sui tipi di eventi supportati, vedere [Eventi](#).

Analisi asincrona per la modellazione degli argomenti

Per determinare gli argomenti in un set di documenti, utilizzate [StartTopicsDetectionJob](#) per avviare un processo asincrono. È possibile monitorare gli argomenti nei documenti scritti in inglese o spagnolo.

Argomenti

- [Prima di iniziare](#)
- [Utilizzando il AWS Command Line Interface](#)
- [Usare l'SDK per Python o SDK per .NET](#)

Prima di iniziare

Prima di iniziare, assicurati di avere:

- Bucket di input e output: identifica i bucket Amazon S3 che desideri utilizzare per input e output. I bucket devono trovarsi nella stessa regione dell'API che stai chiamando.
- Ruolo di servizio IAM: è necessario disporre di un ruolo di servizio IAM con l'autorizzazione ad accedere ai bucket di input e output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Utilizzando il AWS Command Line Interface

Nell'esempio seguente viene illustrato l'utilizzo dell'`StartTopicsDetectionJob` operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```
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
```

Per il `cli-input-json` parametro fornite il percorso di un file JSON che contiene i dati della richiesta, come illustrato nell'esempio seguente.

```
{  
  "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 la richiesta di avvio del processo di rilevamento degli argomenti ha avuto esito positivo, riceverai la seguente risposta:

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

Utilizza l'[ListTopicsDetectionJobs](#) operazione per visualizzare un elenco dei lavori di rilevamento degli argomenti che hai inviato. L'elenco include informazioni sulle posizioni di input e output utilizzate e sullo stato di ciascun processo di rilevamento. L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).


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

In risposta otterrete un codice JSON simile al seguente:

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

È possibile utilizzare l'[DescribeTopicsDetectionJob](#) operazione per ottenere lo stato di un lavoro esistente. L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

In risposta riceverai il seguente codice 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
  }
}
```

Usare l'SDK per Python o SDK per .NET

Per esempi SDK su come avviare un lavoro di modellazione di argomenti, consulta. [Utilizzo StartTopicsDetectionJob con un AWS SDK o una CLI](#)

Fiducia e sicurezza

Gli utenti generano grandi quantità di contenuti testuali tramite applicazioni online (come peer-to-peer chat e discussioni nei forum), commenti pubblicati su siti Web e applicazioni di intelligenza artificiale generativa (richieste di input e output da modelli di intelligenza artificiale generativa). Le funzionalità Amazon Comprehend Trust and Safety possono aiutarti a moderare questi contenuti, per fornire un ambiente sicuro e inclusivo per i tuoi utenti.

I vantaggi dell'utilizzo delle funzionalità di affidabilità e sicurezza di Amazon Comprehend includono:

- **Moderazione più rapida:** modera in modo rapido e preciso grandi volumi di testo per mantenere le piattaforme online libere da contenuti inappropriati.
- **Personalizzabile:** personalizza le soglie di moderazione nelle risposte API per adattarle alle esigenze delle tue applicazioni.
- **Facile da usare:** configura le funzionalità di affidabilità e sicurezza tramite l' LangChain integrazione o l'utilizzo della sala operatoria. AWS CLI SDKs

L'affidabilità e la sicurezza di Amazon Comprehend riguardano i seguenti aspetti della moderazione dei contenuti:

- **Toxicity detection**— Rileva contenuti che potrebbero essere dannosi, offensivi o inappropriati. Gli esempi includono l'incitamento all'odio, le minacce o gli abusi.
- **Intent classification**— Rileva contenuti con intenzioni malevole esplicite o implicite. Gli esempi includono contenuti discriminatori o illegali o contenuti che esprimono o richiedono consulenza su argomenti medici, legali, politici, controversi, personali o finanziari.
- **Privacy protection**— Gli utenti possono fornire inavvertitamente contenuti che possono rivelare informazioni di identificazione personale (PII). Amazon Comprehend PII offre la possibilità di rilevare e redigere informazioni personali.

Argomenti

- [Rilevamento della tossicità](#)
- [Classificazione rapida della sicurezza](#)
- [Rilevamento e redazione delle PII](#)

Rilevamento della tossicità

Il rilevamento della tossicità di Amazon Comprehend fornisce il rilevamento in tempo reale del contenuto tossico nelle interazioni basate su testo. Puoi utilizzare il rilevamento della tossicità per moderare peer-to-peer le conversazioni nelle piattaforme online o per monitorare input e output generativi dell'IA.

Il rilevamento della tossicità rileva le seguenti categorie di contenuti offensivi:

GRAPHIC

Il linguaggio grafico utilizza immagini visivamente descrittive, dettagliate e sgradevolmente vivide. Tale linguaggio viene spesso reso prolisso per amplificare un insulto, un disagio o un danno al destinatario.

MOLESTIE O ABUSI

Un discorso che impone dinamiche di potere dirompenti tra chi parla e chi ascolta, indipendentemente dall'intenzione, cerca di influire sul benessere psicologico del destinatario o oggettivizza una persona.

INCITAMENTO ALL'ODIO

Discorso che critica, insulta, denuncia o disumanizza una persona o un gruppo sulla base di un'identità, che si tratti di razza, etnia, identità di genere, religione, orientamento sessuale, abilità, origine nazionale o un altro gruppo identitario.

INSULTO

Discorso che include un linguaggio umiliante, derisorio, offensivo o sminuente.

PROFANITÀ

I discorsi che contengono parole, frasi o acronimi maleducati, volgari o offensivi sono considerati profani.

SESSUALE

Discorso che indica interesse, attività o eccitazione sessuale utilizzando riferimenti diretti o indiretti a parti del corpo, tratti fisici o sesso.

VIOLENZA_O_MINACCIA

Discorso che include minacce che cercano di infliggere dolore, lesioni o ostilità verso una persona o un gruppo.

TOSSICITÀ

Discorso che contiene parole, frasi o acronimi che potrebbero essere considerati di natura tossica in una delle categorie precedenti.

Rilevamento di contenuti tossici tramite l'API

Per rilevare il contenuto tossico nel testo, utilizzate l'operazione [DetectToxicContents](#) sincrona. Questa operazione esegue l'analisi su un elenco di stringhe di testo fornite come input. La risposta dell'API contiene un elenco di risultati che corrisponde alla dimensione dell'elenco di input.

Attualmente, il rilevamento dei contenuti tossici supporta solo la lingua inglese. Per il testo di input, puoi fornire un elenco di un massimo di 10 stringhe di testo. Ogni stringa ha una dimensione massima di 1 KB.

Il rilevamento del contenuto tossico restituisce un elenco di risultati di analisi, una voce nell'elenco per ogni stringa di input. Una voce contiene un elenco di tipi di contenuto tossico identificati nella stringa di testo, insieme a un punteggio di affidabilità per ogni tipo di contenuto. La voce include anche un punteggio di tossicità per la stringa.

Gli esempi seguenti mostrano come utilizzare l'DetectToxicContent operazione utilizzando AWS CLI and Python.

AWS CLI

È possibile rilevare il contenuto tossico utilizzando il seguente comando in: AWS CLI

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

AWS CLI Risponde con il seguente risultato. Il segmento di testo riceve un punteggio di confidenza elevato nella INSULT categoria, con un conseguente punteggio di tossicità elevato:

```
{  
  "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
}
]
}

```

È possibile inserire fino a 10 stringhe di testo, utilizzando il seguente formato per il `text-segments` parametro:

```

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

```

AWS CLI Risponde con i seguenti risultati:

```
{
```

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

Python (Boto)

L'esempio seguente dimostra come rilevare contenuti tossici usando 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)
```

Classificazione rapida della sicurezza

Amazon Comprehend fornisce un classificatore binario pre-addestrato per classificare le istruzioni di input in testo semplice per modelli di linguaggio di grandi dimensioni (LLM) o altri modelli di intelligenza artificiale generativa.

Il prompt safety classifier analizza il prompt di input e assegna un punteggio di confidenza alla sicurezza o meno del prompt.

Un prompt non sicuro è un prompt di input che esprime intenzioni malevole, ad esempio la richiesta di informazioni personali o private, la generazione di contenuti offensivi o illegali o la richiesta di consulenza su argomenti medici, legali, politici o finanziari.

Classificazione rapida della sicurezza tramite l'API

Per eseguire la classificazione di sicurezza rapida per una stringa di testo, utilizzate l'operazione [ClassifyDocuments](#) sincrona. Per l'input, si fornisce una stringa di testo semplice in inglese. La dimensione massima della stringa è di 10 KB.

La risposta include due classi (SAFE e UNSAFE), oltre a un punteggio di confidenza per ogni classe. L'intervallo di valori del punteggio è compreso tra zero e uno corrisponde al livello di confidenza più elevato.

I seguenti esempi mostrano come utilizzare la classificazione di sicurezza rapida con AWS CLI e Python.

AWS CLI

L'esempio seguente mostra come utilizzare il classificatore di sicurezza prompt con: 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.'
```

AWS CLI Risponde con il seguente risultato:

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


 Note

Quando si utilizza il `classify-document` comando, per il `--endpoint-arn` parametro è necessario passare un ARN che utilizzi lo stesso della configurazione AWS CLI . Per configurare AWS CLI, esegui il `aws configure` comando. In questo esempio, l'ARN dell'endpoint ha il codice `Region. us-west-2` È possibile utilizzare il classificatore di sicurezza prompt in una delle seguenti regioni:

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

Python (Boto)

L'esempio seguente dimostra come utilizzare il classificatore di sicurezza prompt con 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

Quando si utilizza il `classify_document` metodo, per l'`EndpointArn` argomento è necessario passare un ARN che utilizzi lo stesso del client SDK Regione AWS boto3. In questo esempio, il client e l'ARN dell'endpoint utilizzano entrambi. `us-west-2` È possibile utilizzare il prompt safety classifier in una delle seguenti regioni:

- `us-east-1`
- `us-west-2`
- `eu-west-1`

- ap-southeast-2

Rilevamento e redazione delle PII

Puoi utilizzare la console Amazon Comprehend o API per rilevare informazioni di identificazione personale (PII) in documenti di testo in inglese o spagnolo. Le PII sono un riferimento testuale ai dati personali che possono identificare un individuo. Gli esempi di PII includono indirizzi, numeri di conto corrente e numeri di telefono.

È possibile rilevare o oscurare le entità PII nel testo. Per rilevare le entità PII, è possibile utilizzare l'analisi in tempo reale o un processo batch asincrono. Per redigere le entità PII, è necessario utilizzare un processo batch asincrono.

Per ulteriori informazioni, consulta [Informazioni personali di identificazione \(PII\)](#).

Informazioni personali di identificazione (PII)

Puoi utilizzare la console Amazon Comprehend o APIs rilevare informazioni di identificazione personale (PII) in documenti di testo in inglese o spagnolo. Le PII sono un riferimento testuale ai dati personali che potrebbero essere utilizzati per identificare un individuo. Gli esempi di PII includono indirizzi, numeri di conto corrente e numeri di telefono.

Con il rilevamento delle informazioni PII, è possibile scegliere di individuare le entità PII o di oscurarle nel testo. Per individuare le entità PII, è possibile utilizzare l'analisi in tempo reale o un processo batch asincrono. Per redigere le entità PII, è necessario utilizzare un processo batch asincrono.

Puoi utilizzare i punti di accesso Amazon S3 Object Lambda per informazioni di identificazione personale (PII) per controllare il recupero dei documenti dal tuo bucket Amazon S3. Puoi controllare l'accesso ai documenti che contengono informazioni personali e cancellare le informazioni di identificazione personale dai documenti. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Lambda a oggetti Amazon S3 per informazioni di identificazione personale \(PII\)](#).

Argomenti

- [Rilevamento delle entità PII](#)
- [Etichettatura delle entità PII](#)
- [Analisi PII in tempo reale \(console\)](#)
- [Lavori di analisi asincrona PII \(Console\)](#)
- [Analisi delle PII in tempo reale \(API\)](#)
- [processi di analisi asincrona \(API\) PII](#)

Rilevamento delle entità PII

Puoi usare Amazon Comprehend per rilevare le entità PII nei documenti di testo in inglese o spagnolo. Un'entità PII è un tipo specifico di informazioni di identificazione personale (PII). Utilizza il rilevamento delle informazioni PII per individuare le entità PII o oscurare le entità PII nel testo.

Argomenti

- [Individua le entità PII](#)
- [Redigi le entità PII](#)
- [Tipi di entità universali PII](#)

- [Tipi di entità PII specifici per paese](#)

Individua le entità PII

Per individuare le entità PII nel testo, è possibile analizzare rapidamente un singolo documento utilizzando l'analisi in tempo reale. È inoltre possibile avviare un processo batch asincrono su una raccolta di documenti.

Puoi utilizzare la console o l'API per l'analisi in tempo reale di un singolo documento. Il testo di input può includere fino a 100 kilobyte di caratteri con codifica UTF-8.

Ad esempio, puoi inviare il seguente testo di input per individuare le entità PII:

Salve Paulo Santos. L'ultimo estratto conto relativo al conto della tua carta di credito 1111-0000-1111-0000 è stato inviato per posta a 123 Any Street, Seattle, WA 98109.

L'output include le informazioni che «Paul Santos» ha il tipo, «1111-0000-1111-0000» ha il tipo e «123 Any StreetNAME, Seattle, WA 98109» ha il tipo. CREDIT_DEBIT_NUMBER ADDRESS

Amazon Comprehend restituisce un elenco di entità PII rilevate, con le seguenti informazioni per ciascuna entità PII:

- Un punteggio che stima la probabilità che l'intervallo di testo rilevato sia il tipo di entità rilevato.
- Il tipo di entità PII.
- La posizione dell'entità PII nel documento, specificata come offset di caratteri per l'inizio e la fine dell'entità.

Ad esempio, il testo di input menzionato in precedenza produce la seguente risposta:

```
{
  "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  
    }  
]  
}
```

Redigi le entità PII

Per oscurare le entità PII nel testo, puoi utilizzare la console o l'API per avviare un processo batch asincrono. Amazon Comprehend restituisce una copia del testo di input con redazioni per ogni entità PII.

Ad esempio, puoi inviare il seguente testo di input per oscurare le entità PII:

Buongiorno Paulo Santos. L'ultimo estratto conto relativo al conto della tua carta di credito 1111-0000-1111-0000 è stato inviato per posta a 123 Any Street, Seattle, WA 98109.

Il file di output include il testo seguente:

Salve *****. L'ultimo estratto conto relativo al conto della tua carta di credito ***** è stato inviato per posta a *** ***** ***** *****.

Tipi di entità universali PII

Alcuni tipi di entità PII sono universali (non specifici per i singoli paesi), come gli indirizzi e-mail e i numeri di carte di credito. Amazon Comprehend rileva i seguenti tipi di entità PII universali:

ADDRESS

Indirizzo fisico, ad esempio “100 Main Street, Anytown, USA” o “Suite 12, Edificio 123”. Un indirizzo può includere una via, un numero civico, una posizione, una città, uno stato, una provincia, una Regione, un CAP, un distretto, un quartiere e altro ancora.

AGE

Età di un individuo, inclusa la quantità e l'unità di tempo. Ad esempio, nella frase «Ho 40 anni», Amazon Comprehend riconosce «40 anni» come età.

AWS_ACCESS_CHIAVE

Un identificatore univoco associato a una chiave di accesso segreta; si utilizzano l'ID della chiave di accesso e la chiave di accesso segreta per firmare le richieste programmatiche AWS in modo crittografico.

AWS_SECRET_CHIAVE

Identificatore univoco associato a una chiave di accesso. Si utilizzano l'ID della chiave di accesso e la chiave di accesso segreta per firmare le AWS richieste programmatiche in modo crittografico.

CREDIT_DEBIT_CVV

Un codice di verifica della carta a tre cifre (CVV) presente sulle carte di credito e debito VISA e Discover. MasterCard Nelle carte di credito o debito American Express, il codice CVV è un codice numerico a 4 cifre.

CREDIT_DEBIT_EXPIRY

Data di scadenza della carta di credito o della carta di debito. Questo numero è in genere composto da quattro cifre ed è spesso formattato come MM/YY. month/year Amazon Comprehend riconosce date di scadenza come 21/01/2021, 01/2021 e gennaio 2021.

NUMERO_CREDIT_DEBIT_

Numeri di carta di credito o di debito. La lunghezza di questi numeri può variare da 13 a 16 cifre. Tuttavia, Amazon Comprehend riconosce i numeri delle carte di credito o di debito anche quando sono presenti solo le ultime quattro cifre.

DATA_ORA

Una data può includere un anno, un mese, un giorno, un giorno della settimana o un'ora del giorno. Ad esempio, Amazon Comprehend riconosce come date «19 gennaio 2020» o «11:00». Amazon Comprehend riconoscerà date parziali, intervalli di date e intervalli di date. Riconoscerà anche decenni, come «gli anni '90».

ID_DRIVER

Numero assegnato alla patente di guida, ovvero un documento ufficiale che consente a una persona di guidare uno o più veicoli motorizzati su una strada pubblica. Il numero di patente di guida è composto da caratteri alfanumerici.

EMAIL

Indirizzo e-mail, ad esempio marymajor@email.com.

INTERNATIONAL_BANK_ACCOUNT_NUMBER

Numero di conto bancario internazionale con formati specifici in ogni paese. Vedi www.iban.com/structure.

IP_ADDRESS

Un IPv4 indirizzo, ad esempio 198.51.100.0.

LICENSE_PLATE

La targa di un veicolo viene emessa dal paese in cui il veicolo è immatricolato. Per i veicoli passeggeri sono in genere utilizzate da cinque a otto cifre, con lettere maiuscole e numeri. Il formato varia a seconda dell'ubicazione del paese di emissione.

MAC_ADDRESS

Un indirizzo MAC (Media Access Control) è un identificatore univoco assegnato a un controller di interfaccia di rete (NIC).

NAME

Nome di una persona. Questo tipo di entità non include titoli, ad esempio Dottore, Dottoressa, Signore o Signora. Amazon Comprehend non applica questo tipo di entità a nomi che fanno parte di organizzazioni o indirizzi. Ad esempio, Amazon Comprehend riconosce la «John Doe Organization» come organizzazione e riconosce «Jane Doe Street» come indirizzo.

PASSWORD

Stringa alfanumerica utilizzata come password, ad esempio “*very20special#pass*”.

PHONE

Numero di telefono. Questo tipo di entità include anche numeri di fax e cercapersone.

PIN

Codice di identificazione personale a quattro cifre (PIN) con cui è possibile accedere al proprio conto bancario.

SWIFT_CODE

Un codice SWIFT è un formato standard di codice BIC (Bank Identifier Code) utilizzato per specificare una particolare banca o filiale. Le banche utilizzano questi codici per i trasferimenti di denaro, come i bonifici internazionali.

I codici SWIFT sono composti da 8 o 11 caratteri. I codici a 11 cifre si riferiscono a filiali specifiche, mentre i codici a 8 cifre (o codici a 11 cifre che terminano con “XXX”) si riferiscono alla sede centrale o principale.

URL

Indirizzo web, ad esempio www.esempio.com.

USERNAME

Nome utente che identifica un account, ad esempio un nome di accesso, un nome utente, un soprannome o un handle.

VEHICLE_IDENTIFICATION_NUMBER

Numero di matricola del veicolo che identifica in modo univoco un veicolo. Il contenuto e il formato di tale numero sono definiti nella specifica ISO 3779. Ogni paese ha codici e formati specifici per VINs

Tipi di entità PII specifici per paese

Alcuni tipi di entità PII sono specifici del paese, come i numeri di passaporto e altri numeri di identità emessi dal governo. Amazon Comprehend rileva i seguenti tipi di entità PII specifiche del paese:

CA_HEALTH_NUMBER

Numero del servizio sanitario canadese, ovvero un identificatore univoco di 10 cifre, necessario per consentire alle persone di accedere alle prestazioni sanitarie.

CA_SOCIAL_INSURANCE_NUMBER

Numero di previdenza sociale (SIN) del Canada, ovvero un identificatore univoco a nove cifre, necessario per consentire alle persone di accedere ai programmi e ai benefici governativi.

Il codice SIN è formattato in tre gruppi di cifre, ad esempio 123-456-789. Un codice SIN può essere convalidato tramite un semplice processo a cifra di controllo denominato [algoritmo Luhn](#).

IN_AADHAAR

Un Aadhaar indiano è un numero di identificazione univoco di 12 cifre rilasciato dal governo indiano ai residenti dell'India. Il formato Aadhaar presenta uno spazio o un trattino dopo la quarta e l'ottava cifra.

IN_REGA

Un numero dell'Indian National Rural Employment Guarantee Act (NREGA) è composto da due lettere seguite da 14 numeri.

NUMERO_CONTO_PERMANENTE

Un numero di conto permanente indiano è un numero alfanumerico univoco di 10 cifre emesso dal Dipartimento delle imposte sul reddito.

IN_VOTER_NUMBER

Un ID elettorale indiano è composto da tre lettere seguite da sette numeri.

UK_NATIONAL_HEALTH_SERVICE_NUMBER

Numero del servizio sanitario del Regno Unito, ovvero un identificatore di 10-17 cifre, ad esempio 485 777 3456. Nel sistema attuale il formato del numero è a 10 cifre con spazi dopo la terza e la sesta cifra. L'ultima cifra è un checksum che rileva gli errori.

Il formato numerico a 17 cifre presenta spazi dopo la decima e la tredicesima cifra.

UK_NATIONAL_INSURANCE_NUMBER

Numero di previdenza sociale (NINO) del Regno Unito, che fornisce alle persone l'accesso alle prestazioni della previdenza sociale. Viene anche utilizzato per alcuni scopi nel sistema fiscale del Regno Unito.

Il numero è composto da nove cifre e inizia con due lettere, seguite da sei numeri e una lettera. Un codice NINO può essere formattato con uno spazio o un trattino dopo le due lettere e dopo la seconda, la quarta e la sesta cifra.

UK_UNIQUE_TAXPAYER_REFERENCE_NUMBER

Codice UTR (Unique Taxpayer Reference) del Regno Unito, ovvero un numero di 10 cifre che identifica un contribuente o un'azienda.

NUMERO_CONTO_BANCARIO

Numero di conto bancario statunitense, composto in genere da 10 a 12 cifre. Amazon Comprehend riconosce anche i numeri di conto bancario quando sono presenti solo le ultime quattro cifre.

BANK_ROUTING

Numero di routing del conto bancario statunitense. In genere sono composti da nove cifre, ma Amazon Comprehend riconosce i numeri di routing anche quando sono presenti solo le ultime quattro cifre.

NUMERO_PASSAPORTO

Numero di passaporto degli Stati Uniti. I numeri dei passaporti sono composti da sei a nove caratteri alfanumerici.

US_INDIVIDUAL_TAX_IDENTIFICATION_NUMBER

Codice identificativo individuale del contribuente statunitense (ITIN), ovvero un numero di nove cifre che inizia con un "9" e contiene un "7" oppure un "8" come quarta cifra. Un codice ITIN può essere formattato con uno spazio o un trattino dopo la terza e la quarta cifra.

SSN

Numero di previdenza sociale (SSN) degli Stati Uniti, ovvero un numero di nove cifre rilasciato ai cittadini statunitensi, ai residenti permanenti e ai lavoratori con residenza temporanea. Amazon Comprehend riconosce anche i numeri di previdenza sociale quando sono presenti solo le ultime quattro cifre.

Etichettatura delle entità PII

Quando esegui il rilevamento delle informazioni PII, Amazon Comprehend restituisce le etichette dei tipi di entità PII identificati. Ad esempio, se invii il seguente testo di input ad Amazon Comprehend:

Salve Paulo Santos. L'ultimo estratto conto relativo al conto della tua carta di credito 1111-0000-1111-0000 è stato inviato per posta a 123 Any Street, Seattle, WA 98109.

L'output include etichette che rappresentano i tipi di entità PII insieme a un punteggio di affidabilità dell'accuratezza. In questo caso, il testo del documento «Paul Santos», «1111-0000-1111-0000» e «123 Any Street, Seattle, WA 98109» generano le etichette e rispettivamente come tipi di entità PII. NAME CREDIT_DEBIT_NUMBER ADDRESS Per [Tipi di entità universali PII](#) ulteriori informazioni sui tipi di entità supportati, vedere.

Amazon Comprehend fornisce le seguenti informazioni per ogni etichetta:

- Il nome dell'etichetta del tipo di entità PII.
- Un punteggio che stima la probabilità che il testo rilevato sia etichettato come tipo di entità PII.

L'esempio di testo di input riportato sopra produce il seguente output JSON.

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

Analisi PII in tempo reale (console)

È possibile utilizzare la console per eseguire il rilevamento delle PII in tempo reale di un documento di testo. La dimensione massima del testo è di 100 kilobyte di caratteri con codifica UTF-8. La console visualizza i risultati in modo che sia possibile rivedere l'analisi.

Esegui l'analisi in tempo reale del rilevamento delle PII utilizzando il modello integrato

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analisi in tempo reale.
3. In Tipo di input, scegli Integrato per il tipo di analisi.
4. Inserisci il testo che desideri analizzare.
5. Scegli Analizza. La console visualizza i risultati dell'analisi del testo nel pannello Insights. La scheda PII elenca le entità PII rilevate nel testo di input.

Nel pannello Insights, la scheda PII mostra i risultati per due modalità di analisi:

- Offset: identifica la posizione delle PII nel documento di testo.
- Etichette: identifica le etichette dei tipi di entità PII identificati.

Offset

La modalità di analisi Offsets identifica la posizione delle PII nei documenti di testo. Per ulteriori informazioni, consulta [Individua le entità 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

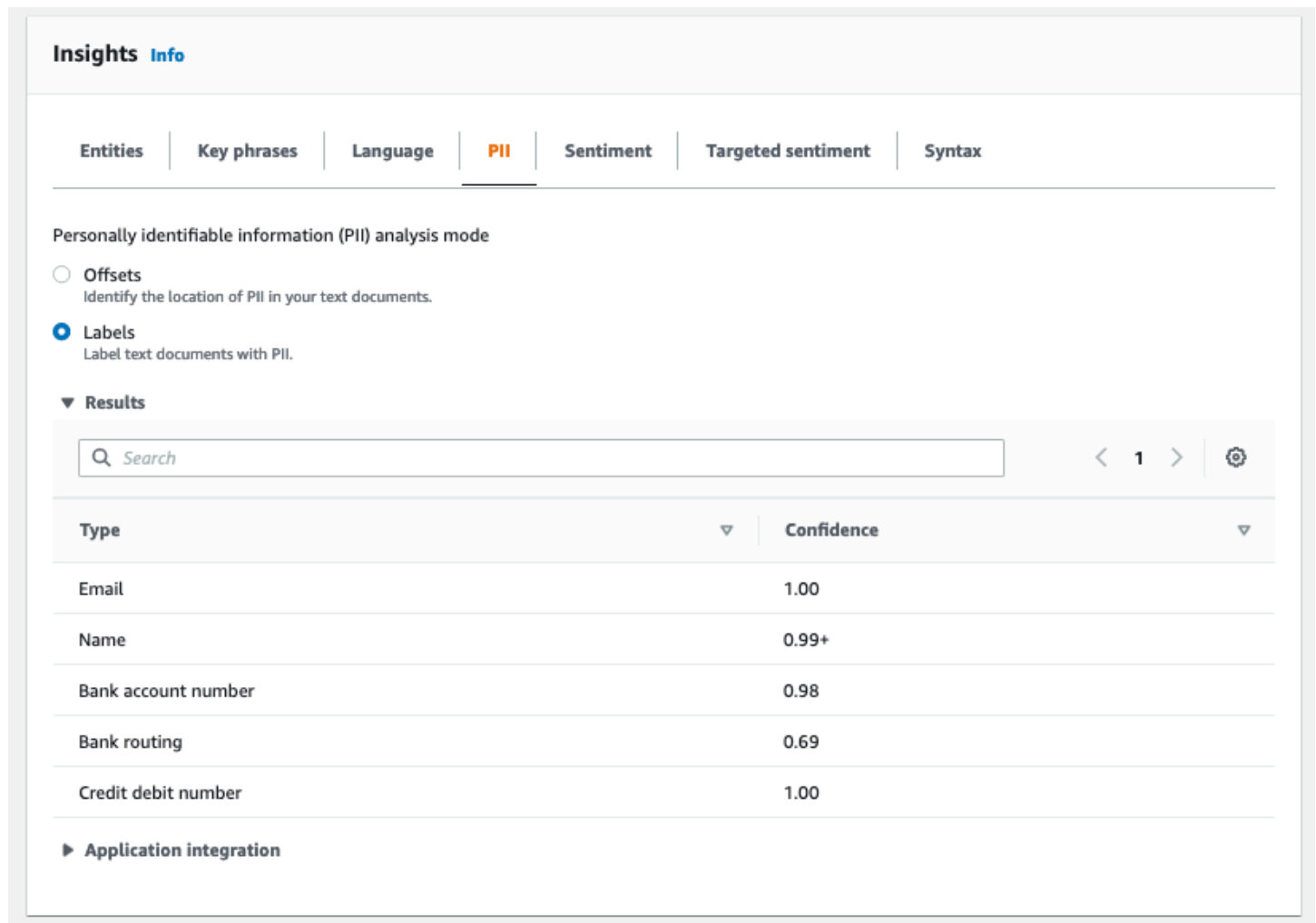
< 1 >

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

Etichette

La modalità di analisi Labels restituisce le etichette dei tipi di entità PII identificati. Per ulteriori informazioni, consulta [Etichettatura delle entità PII](#).



The screenshot shows the Amazon Comprehend Insights console interface. At the top, there's a header with 'Insights Info'. Below it, a navigation bar includes tabs for 'Entities', 'Key phrases', 'Language', 'PII' (which is selected and highlighted in orange), 'Sentiment', 'Targeted sentiment', and 'Syntax'. The main content area is titled 'Personally identifiable information (PII) analysis mode'. It features two radio buttons: 'Offsets' (unselected) and 'Labels' (selected). Below these, a 'Results' section is visible, containing a search bar and a table of results. The table has two columns: 'Type' and 'Confidence'. The results list five types of PII with their respective confidence scores: Email (1.00), Name (0.99+), Bank account number (0.98), Bank routing (0.69), and Credit debit number (1.00). At the bottom of the console, there is a link for 'Application integration'.

Type	Confidence
Email	1.00
Name	0.99+
Bank account number	0.98
Bank routing	0.69
Credit debit number	1.00

Lavori di analisi asincrona PII (Console)

È possibile utilizzare la console per creare processi di analisi asincroni per rilevare entità PII. Per ulteriori informazioni sui tipi di entità PII, vedere [Rilevamento delle entità PII](#)

Per creare un processo di analisi

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analysis jobs, quindi scegli Crea job.

3. In Impostazioni Job, assegnate al job di analisi un nome univoco.
4. Per Tipo di analisi, scegli Informazioni di identificazione personale (PII).
5. Per Lingua, scegli una delle lingue supportate (inglese o spagnolo).
6. Dalla modalità Output, selezionate una delle seguenti opzioni:
 - Offset: l'output del lavoro restituisce la posizione di ogni entità PII.
 - Redazioni: l'output del lavoro restituisce una copia del testo di input con ogni voce PII oscurata.
7. (Facoltativo) Se scegliete Redazioni come modalità di output, potete selezionare i tipi di entità PII da oscurare.
8. In Dati di input, specifica dove si trovano i documenti di input in Amazon S3:
 - Per analizzare i tuoi documenti, scegli I miei documenti e scegli Sfoglia S3 per fornire il percorso del bucket o della cartella che contiene i tuoi file.
 - Per analizzare i campioni forniti da Amazon Comprehend, scegli Documenti di esempio. In questo caso, Amazon Comprehend utilizza un bucket gestito da AWS e non viene specificata la posizione.
9. (Facoltativo) Per il formato di input, specifica uno dei seguenti formati per i tuoi file di input:
 - Un documento per file: ogni file contiene un documento di input. Questa soluzione è ideale per raccolte di documenti di grandi dimensioni.
 - Un documento per riga: l'input è costituito da uno o più file. Ogni riga di un file è considerata un documento. Questa opzione è ideale per documenti brevi, come i post sui social media. Ogni riga deve terminare con un'alimentazione di linea (LF,\n), un valore di ritorno (CR,\r) o entrambi (CRLF,\r\n). Non è possibile utilizzare il separatore di riga UTF-8 (u+2028) per terminare una riga.
10. In Dati di output, scegli Browse S3. Scegli il bucket o la cartella Amazon S3 in cui desideri che Amazon Comprehend scriva i dati di output prodotti dall'analisi.
11. (Facoltativo) Per crittografare il risultato dell'output del tuo lavoro, scegli Crittografia. Quindi, scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account:
 - Se utilizzi una chiave associata all'account corrente, scegli l'alias o l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'alias o l'ID della chiave in KMS key ID.

 Note

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta [Servizio di gestione delle chiavi \(KMS\)](#).

12. In Autorizzazioni di accesso, fornisci un ruolo IAM che:

- Garantisce l'accesso in lettura alla posizione Amazon S3 dei documenti di input.
- Concede l'accesso in scrittura alla posizione Amazon S3 dei documenti di output.
- Include una politica di fiducia che consente al responsabile del comprehend.amazonaws.com servizio di assumere il ruolo e ottenere le relative autorizzazioni.

Se non disponi già di un ruolo IAM con queste autorizzazioni e una policy di fiducia appropriata, scegli Crea un ruolo IAM per crearne uno.

13. Quando hai finito di compilare il modulo, scegli Crea lavoro per creare e avviare il processo di rilevamento degli argomenti.

Il nuovo lavoro viene visualizzato nell'elenco dei lavori con il campo di stato che mostra lo stato del lavoro. Il campo può essere IN_PROGRESS relativo a un lavoro in fase COMPLETED di elaborazione, a un lavoro terminato correttamente e FAILED a un lavoro con un errore. È possibile fare clic su un lavoro per ottenere ulteriori informazioni sul lavoro, inclusi eventuali messaggi di errore.

Una volta completato il processo, Amazon Comprehend archivia i risultati dell'analisi nella posizione di output Amazon S3 specificata per il lavoro. Per una descrizione dei risultati dell'analisi, consulta.

[Rilevamento delle entità PII](#)

Analisi delle PII in tempo reale (API)

Amazon Comprehend fornisce operazioni API sincrone in tempo reale per analizzare le informazioni di identificazione personale (PII) in un documento.

Argomenti

- [Individuazione delle entità PII in tempo reale \(API\)](#)
- [Etichettatura delle entità PII in tempo reale \(API\)](#)

Individuazione delle entità PII in tempo reale (API)

Per individuare le informazioni personali in un unico documento, puoi utilizzare l'operazione Amazon [DetectPiiEntities](#) Comprehend. Il testo di input può includere fino a 100 kilobyte di caratteri codificati UTF-8. Le lingue supportate includono inglese e spagnolo.

Individuazione delle PII tramite (CLI)

L'esempio seguente utilizza l'`DetectPiiEntities` operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

Amazon Comprehend risponde con quanto segue:

```
{  
  "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  
    }  
  ]  
}
```



```
]
}
```

Etichettatura delle entità PII in tempo reale (API)

È possibile utilizzare operazioni API sincrone in tempo reale per restituire le etichette dei tipi di entità PII identificati. Per ulteriori informazioni, consulta [Etichettatura delle entità PII](#).

Etichettatura delle entità PII (CLI)

L'esempio seguente utilizza l'`ContainsPiiEntities` operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

Amazon Comprehend risponde con quanto segue:

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

processi di analisi asincrona (API) PII

Analisi asincrona PII (API)

È possibile utilizzare operazioni API asincrone per creare processi di analisi per individuare o redigere entità PII. Per ulteriori informazioni sui tipi di entità PII, vedere. [Rilevamento delle entità PII](#)

Argomenti

- [Individuazione di entità PII con processi asincroni \(API\)](#)
- [Oscurazione delle entità PII con processi asincroni \(API\)](#)

Individuazione di entità PII con processi asincroni (API)

Esegui un processo batch asincrono per individuare le PII in una raccolta di documenti. Per eseguire il lavoro, carica i tuoi documenti su Amazon S3 e invia una [StartPiiEntitiesDetectionJob](#) richiesta.

Argomenti

- [Prima di iniziare](#)
- [Parametri di input](#)
- [Metodi Async Job](#)
- [Formato del file di output](#)
- [Analisi asincrona utilizzando il AWS Command Line Interface](#)

Prima di iniziare

Prima di iniziare, assicurati di avere:

- Bucket di input e output: identifica i bucket Amazon S3 che desideri utilizzare per file di input e file di output. I bucket devono trovarsi nella stessa regione dell'API che stai chiamando.
- Ruolo di servizio IAM: è necessario disporre di un ruolo di servizio IAM con l'autorizzazione ad accedere ai bucket di input e output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Parametri di input

Nella richiesta, includi i seguenti parametri obbligatori:

- **InputDataConfig**— Fornite una [InputDataConfig](#) definizione per la richiesta, che include le proprietà di input per il lavoro. Per il **S3Uri** parametro, specifica la posizione in Amazon S3 dei documenti di input.
- **OutputDataConfig**— Fornisci una [OutputDataConfig](#) definizione per la tua richiesta, che include le proprietà di output per il lavoro. Per il **S3Uri** parametro, specifica la posizione Amazon S3 in cui Amazon Comprehend scrive i risultati della sua analisi.
- **DataAccessRoleArn**— Fornisci l'Amazon Resource Name (ARN) di un AWS Identity and Access Management ruolo. Questo ruolo deve concedere ad Amazon Comprehend l'accesso in lettura ai dati di input e scrittura alla posizione di output in Amazon S3. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).
- **Mode**— Imposta questo parametro su **ONLY_OFFSETS**. Con questa impostazione, l'output fornisce gli offset di caratteri che individuano ogni entità PII nel testo di input. L'output include anche punteggi di confidenza e tipi di entità PII.
- **LanguageCode**— Impostate questo parametro su **en o es**. Amazon Comprehend supporta il rilevamento delle informazioni PII in testo in inglese o spagnolo.

Metodi Async Job

StartPiiEntitiesDetectionJob Restituisce un ID del lavoro, in modo da poter monitorare l'avanzamento del lavoro e recuperare lo stato del lavoro una volta completato.

Per monitorare lo stato di avanzamento di un processo di analisi, fornite l'ID del lavoro all'[DescribePiiEntitiesDetectionJob](#) operazione. Il modulo di risposta **DescribePiiEntitiesDetectionJob** contiene il **JobStatus** campo con lo stato corrente del lavoro. Un lavoro di successo passa attraverso i seguenti stati:

INVIATO -> IN_PROGRESS -> COMPLETATO.

Al termine di un processo di analisi (**JobStatus** completato, non riuscito o interrotto), utilizzatelo **DescribePiiEntitiesDetectionJob** per ottenere la posizione dei risultati. Se lo stato del lavoro è **COMPLETED**, la risposta include un **OutputDataConfig** campo che contiene un campo con la posizione Amazon S3 del file di output.

Per ulteriori dettagli sui passaggi da seguire per l'analisi asincrona di Amazon Comprehend, consulta [Elaborazione asincrona in batch](#)

Formato del file di output

Il file di output utilizza il nome del file di input, con .out aggiunto alla fine. Contiene i risultati dell'analisi.

Di seguito è riportato un esempio di file di output di un processo di analisi che ha rilevato entità PII nei documenti. Il formato dell'input è un documento per riga.

```
{
  "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
}
```

Di seguito è riportato un esempio di output di un'analisi in cui il formato dell'input è un documento per file.

```
{
  "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"
}
```

Analisi asincrona utilizzando il AWS Command Line Interface

L'esempio seguente utilizza l'`StartPiiEntitiesDetectionJob` operazione con AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

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

Per il `cli-input-json` parametro si fornisce il percorso di un file JSON che contiene i dati della richiesta, come illustrato nell'esempio seguente.

```
{  
  "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"  
}
```

Se la richiesta di avvio del processo di rilevamento degli eventi ha avuto esito positivo, riceverai una risposta simile alla seguente:

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

È possibile utilizzare l'[DescribeEventsDetectionJob](#) operazione per ottenere lo stato di un lavoro esistente. Se la richiesta di avvio del processo di rilevamento degli eventi ha avuto esito positivo, riceverai una risposta simile alla seguente:

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

Quando il processo viene completato correttamente, riceverai una risposta simile alla seguente:

```
{  
  "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)  
    }  
  }  
}
```

Oscurazione delle entità PII con processi asincroni (API)

Per oscurare le entità PII nel testo, si avvia un processo batch asincrono. Per eseguire il lavoro, carica i tuoi documenti su Amazon S3 e invia una [StartPiiEntitiesDetectionJob](#) richiesta.

Argomenti

- [Prima di iniziare](#)
- [Parametri di input](#)
- [Formato del file di output](#)
- [Redazione delle informazioni PII utilizzando il AWS Command Line Interface](#)

Prima di iniziare

Prima di iniziare, assicurati di avere:

- Bucket di input e output: identifica i bucket Amazon S3 che desideri utilizzare per file di input e file di output. I bucket devono trovarsi nella stessa regione dell'API che stai chiamando.

- Ruolo di servizio IAM: è necessario disporre di un ruolo di servizio IAM con l'autorizzazione ad accedere ai bucket di input e output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Parametri di input

Nella richiesta, includi i seguenti parametri obbligatori:

- **InputDataConfig**— Fornite una [InputDataConfig](#) definizione per la richiesta, che include le proprietà di input per il lavoro. Per il `S3Uri` parametro, specifica la posizione in Amazon S3 dei documenti di input.
- **OutputDataConfig**— Fornisci una [OutputDataConfig](#) definizione per la tua richiesta, che include le proprietà di output per il lavoro. Per il `S3Uri` parametro, specifica la posizione Amazon S3 in cui Amazon Comprehend scrive i risultati della sua analisi.
- **DataAccessRoleArn**— Fornisci l'Amazon Resource Name (ARN) di un AWS Identity and Access Management ruolo. Questo ruolo deve concedere ad Amazon Comprehend l'accesso in lettura ai dati di input e scrittura alla posizione di output in Amazon S3. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).
- **Mode**— Imposta questo parametro su `ONLY_REDACTION`. Con questa impostazione, Amazon Comprehend scrive una copia dei documenti di input nella posizione di output in Amazon S3. In questa copia, ogni entità PII viene redatta.
- **RedactionConfig**— Fornisci una [RedactionConfig](#) definizione per la tua richiesta, che include i parametri di configurazione per la redazione. Specificate i tipi di PII da oscurare e specificate se ogni entità PII viene sostituita con il nome del relativo tipo o con un carattere a scelta:
 - Specificate i tipi di entità PII da oscurare nell'array. `PiiEntityTypes` Per oscurare tutti i tipi di entità, imposta il valore dell'array su `["ALL"]`
 - Per sostituire ogni entità PII con il relativo tipo, imposta il `MaskMode` parametro su `REPLACE_WITH_PII_ENTITY_TYPE`. Ad esempio, con questa impostazione, l'entità PII «Jane Doe» viene sostituita da «[NAME]».
 - Per sostituire i caratteri in ogni entità PII con un carattere a tua scelta, imposta il `MaskMode` parametro su `REPLACE_WITH_MASK_CHARACTER` e imposta il `MaskCharacter` parametro sul carattere sostitutivo. `MASK` Fornisci un solo carattere. I caratteri validi sono `! , #, $, %, &, * e @`. Ad esempio, con questa impostazione, l'entità PII «Jane Doe» può essere sostituita con «**** ***»
- **LanguageCode**— Impostate questo parametro su `en` o `es`. Amazon Comprehend supporta il rilevamento delle informazioni PII in testo in inglese o spagnolo.

Formato del file di output

L'esempio seguente mostra i file di input e output di un processo di analisi che redige le PII. Il formato dell'input è un documento per riga.

```
{
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
}
```

Il lavoro di analisi per redigere questo file di input produce il seguente file di output.

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

Redazione delle informazioni PII utilizzando il AWS Command Line Interface

L'esempio seguente utilizza l'StartPiiEntitiesDetectionJoboperazione con. AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

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

Per il `cli-input-json` parametro si fornisce il percorso di un file JSON che contiene i dati della richiesta, come illustrato nell'esempio seguente.

```
{
  "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"]
    }
}

```

Se la richiesta di avvio del processo di rilevamento degli eventi ha avuto esito positivo, riceverai una risposta simile alla seguente:

```

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

```

È possibile utilizzare l'[DescribeEventsDetectionJob](#) operazione per ottenere lo stato di un lavoro esistente.

```

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

```

Quando il processo viene completato correttamente, si riceve una risposta simile alla seguente:

```

{
  "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)  
}
```

Elaborazione dei documenti

Amazon Comprehend supporta l'elaborazione di documenti in un'unica fase per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Ad esempio, è possibile inserire una combinazione di documenti di testo semplice e documenti semistrutturati (come documenti PDF, documenti Microsoft Word e immagini) in un lavoro di analisi personalizzato.

Per i file di input che richiedono l'estrazione del testo, Amazon Comprehend esegue automaticamente l'estrazione del testo prima di eseguire l'analisi. Per estrarre il contenuto del testo, Amazon Comprehend utilizza un parser interno per documenti semi-strutturati nativi e utilizza Amazon Textract per immagini e documenti API scansionati.

L'elaborazione dei documenti di Amazon Comprehend è disponibile in tutti gli Amazon [Regioni supportate](#) Comprehend, ad eccezione di Asia Pacifico (Tokyo) AWS GovCloud e (Stati Uniti occidentali) che supportano solo modelli di testo semplice per la classificazione personalizzata.

I seguenti argomenti forniscono dettagli sui tipi di documenti di input supportati da Amazon Comprehend per l'analisi personalizzata.

Argomenti

- [Ingressi per analisi personalizzate in tempo reale](#)
- [Ingressi per analisi asincrone personalizzate](#)
- [Impostazione delle opzioni di estrazione del testo](#)
- [Procedure consigliate per le immagini](#)

Ingressi per analisi personalizzate in tempo reale

L'analisi in tempo reale utilizzando modelli personalizzati richiede un singolo documento come input. I seguenti argomenti descrivono i tipi di documenti di input che è possibile utilizzare.

Argomenti

- [Documenti in testo semplice](#)
- [Documenti semistrutturati](#)
- [File di immagine e file PDF scansionati](#)
- [Output di Amazon Textract](#)

- [Dimensioni massime dei documenti per l'analisi in tempo reale](#)
- [Errori nei documenti semistrutturati](#)

Documenti in testo semplice

Fornisci il documento di input come testo in formato UTF-8.

Documenti semistrutturati

I documenti semistrutturati includono documenti PDF nativi e documenti Word.

Per impostazione predefinita, l'analisi personalizzata in tempo reale utilizza il parser Amazon Comprehend per estrarre il testo da file Word e file PDF digitali. Per i file PDF, puoi ignorare questa impostazione predefinita e utilizzare Amazon Textract per estrarre il testo. Per informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).

File di immagine e file PDF scansionati

I tipi di immagini supportati includono JPEG, PNG e TIFF.

Per impostazione predefinita, il riconoscimento personalizzato delle entità utilizza l'operazione DetectDocumentText API Amazon Textract per estrarre il testo dai file di immagine e dai file PDF scansionati. Puoi sovrascrivere questa impostazione predefinita per utilizzare invece l'operazione AnalyzeDocument API. Per informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).

Output di Amazon Textract

Puoi fornire l'output JSON dell'API AnalyzeDocument o dell'API Amazon DetectDocumentText Textract come input per le operazioni API in tempo reale per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Amazon Comprehend supporta questo tipo di input per le operazioni API in tempo reale, ma non per la console.

Dimensioni massime dei documenti per l'analisi in tempo reale

Per tutti i tipi di documenti di input, il file di input può contenere al massimo una pagina, con non più di 10.000 caratteri.

La tabella seguente mostra le dimensioni massime dei file per i documenti di input.

Tipo di file	Dimensione massima (API)	Dimensione massima (console)
Documenti di testo UTF-8	10 KB	10 KB
Documenti PDF	10 MB	5 MB
Documenti Word	10 MB	1 MB
File di immagine	10 MB	5 MB
Textract dei file di output	1 MB	N/A

Errori nei documenti semistrutturati

L'operazione [ClassifyDocument](#) [DetectEntities](#) API può riscontrare errori a livello di documento o a livello di pagina durante l'estrazione del testo da un documento semistrutturato o da un file di immagine.

Errori a livello di pagina

Se l'operazione [ClassifyDocument](#) o l'[DetectEntities](#) API rileva errori durante l'elaborazione di una pagina nel documento di input, la risposta API include una voce nell'[elenco Errori](#) per ogni errore.

La voce `ErrorCode` nell'elenco degli errori contiene uno dei seguenti valori:

- `TEXTTRACT_BAD_PAGE` — Amazon Textract non è in grado di leggere la pagina. Per ulteriori informazioni sui limiti di pagina in Amazon Textract, consulta [Quotazioni di pagina in Amazon Textract](#).
- `TEXTTRACT_PROVISIONED_THROUGHPUT_EXCEEDED` — Il numero di richieste ha superato il limite di velocità effettiva. Per ulteriori informazioni sulle quote di velocità effettiva in Amazon Textract, [consulta Quote predefinite in](#) Amazon Textract.
- `PAGE_CHARACTERS_EXCEEDED` — Troppi caratteri di testo nella pagina (massimo 10.000 caratteri).
- `PAGE_SIZE_EXCEEDED`: la dimensione massima della pagina è 10 MB.
- `INTERNAL_SERVER_ERROR` — La richiesta ha rilevato un problema di servizio. Prova di nuovo la richiesta API.

Errori a livello di documento

Se l'operazione [ClassifyDocument](#) o l'[DetectEntities](#) API rileva un errore a livello di documento nel documento di input, l'API restituisce una risposta di errore. `InvalidRequestException`

Nella risposta all'errore, il Reason campo contiene il valore. `INVALID_DOCUMENT`

Il Detail campo contiene uno dei seguenti valori:

- `DOCUMENT_SIZE_EXCEEDED` — La dimensione del documento è troppo grande. Controlla le dimensioni del file e invia nuovamente la richiesta.
- `UNSUPPORTED_DOC_TYPE` — Il tipo di documento non è supportato. Controlla il tipo di file e invia nuovamente la richiesta.
- `PAGE_LIMIT_EXCEEDED` — Troppe pagine nel documento. Controlla il numero di pagine del file e invia nuovamente la richiesta.
- `TEXTTRACT_ACCESS_DENIED_EXCEPTION` — Accesso negato ad Amazon Textract. Verifica che il tuo account sia autorizzato a utilizzare le operazioni Amazon Textract [DetectDocumentText](#) e [AnalyzeDocument](#) API e invia nuovamente la richiesta.

Ingressi per analisi asincrone personalizzate

È possibile inserire più documenti in un processo di analisi asincrona personalizzato. I seguenti argomenti descrivono i tipi di documenti di input che è possibile utilizzare. La dimensione massima del file varia a seconda del tipo di documento di input.

Argomenti

- [Documenti in testo semplice](#)
- [Documenti semistrutturati](#)
- [File di immagine e file PDF scansionati](#)
- [File JSON di output di Amazon Textract](#)

Documenti in testo semplice

Fornisci tutti i documenti di input in testo normale come testo in formato UTF-8. La tabella seguente elenca le dimensioni massime dei file e altre linee guida.

Note

Questi limiti si applicano quando tutti i file di input sono in testo semplice.

Description	Quota/Linea guida
Dimensione massima del file per un documento per formato di file (classificazione personalizzata)	1 byte: 10 MB
Dimensione del documento (riconoscimento personalizzato delle entità)	1 byte—1 MB
Numero massimo di file, un documento per file	1.000.000
Numero massimo di righe, un documento per riga (per tutti i file richiesti)	1.000.000
Dimensione del corpus del documento (tutti i documenti in testo semplice combinati)	1 byte: 5 GB

Documenti semistrutturati

I documenti semistrutturati includono documenti PDF nativi e documenti Word.

La tabella seguente elenca le dimensioni massime dei file e altre linee guida.

Description	Quota/Linea guida
Dimensioni del documento (PDF)	1 byte: 50 MB
Dimensioni del documento (Docx)	1 byte: 5 MB
Numero massimo di file	500
Numero massimo di pagine per un file PDF o Docx	100

Description	Quota/Linea guida
Dimensione del corpus del documento dopo l'estrazione del testo (testo semplice, tutti i file combinati)	1 byte: 5 GB

Per impostazione predefinita, l'analisi personalizzata utilizza il parser Amazon Comprehend per estrarre il testo da file Word e file PDF digitali. Per i file PDF, puoi ignorare questa impostazione predefinita e utilizzare Amazon Textract per estrarre il testo. Per informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).

File di immagine e file PDF scansionati

L'analisi personalizzata supporta immagini JPEG, PNG e TIFF.

La tabella seguente elenca le dimensioni massime dei file per le immagini. I file PDF scansionati sono soggetti alle stesse dimensioni massime dei file PDF nativi.

Description	Quota/Linea guida
Dimensioni dell'immagine (JPG o PNG)	1 byte—10 MB
Dimensione dell'immagine (TIFF)	1 byte: 10 MB. Massimo una pagina.

Per ulteriori informazioni sulle immagini, vedere [Procedure consigliate per le immagini](#).

Per impostazione predefinita, Amazon Comprehend utilizza l'operazione API Amazon DetectDocumentText Textract per estrarre il testo dai file di immagine e dai file PDF scansionati. Puoi sovrascrivere questa impostazione predefinita per utilizzare invece l'AnalyzeDocument operazione API. Per informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).

File JSON di output di Amazon Textract

Per il riconoscimento personalizzato delle entità, ma non per la classificazione personalizzata, puoi fornire il file di output dell'operazione AnalyzeDocument API Amazon Textract come input per i lavori di analisi.

Impostazione delle opzioni di estrazione del testo

Per impostazione predefinita, Amazon Comprehend esegue le seguenti azioni per estrarre testo da un file, in base al tipo di file di input:

- File Word: il parser Amazon Comprehend estrae il testo.
- File PDF digitali: il parser Amazon Comprehend estrae il testo.
- File di immagine e file PDF scansionati: Amazon Comprehend utilizza l'API Amazon Textract per `DetectDocumentText` estrarre il testo.

Per i file di immagine e i file PDF, puoi utilizzare il `DocumentReaderConfig` parametro per sovrascrivere queste azioni di estrazione predefinite. Questo parametro è disponibile quando utilizzi la console o l'API Amazon Comprehend per analisi personalizzate in tempo reale o asincrone.

Il `DocumentReaderConfig` parametro contiene tre campi:

- `DocumentReadMode`— Impostato `SERVICE_DEFAULT` per consentire ad Amazon Comprehend di eseguire le azioni predefinite.

Imposta `FORCE_DOCUMENT_READ_ACTION` per utilizzare Amazon Textract per analizzare file PDF digitali.

- `DocumentReadAction`— Imposta l'API Amazon Textract (`DetectDocumentText` o `AnalyzeDocument`) da utilizzare quando Amazon Comprehend utilizza Amazon Textract per l'estrazione del testo.
- `FeatureTypes`— Se decidi `DocumentReadAction` di utilizzare l'operazione `AnalyzeDocument` API, puoi aggiungere una o entrambe le `FeatureTypes` (`TABLES`, `FORMS`). Queste funzionalità forniscono informazioni aggiuntive sulle tabelle e sui moduli del documento. Per ulteriori informazioni su queste funzionalità, consulta [Amazon Textract Document Analysis Response Objects](#).

I seguenti esempi mostrano come configurare `DocumentReaderConfig` per casi d'uso specifici:

1. Usa Amazon Textract per tutti i file PDF.
 - a. `DocumentReadMode`: impostato su `FORCE_DOCUMENT_READ_ACTION`.
 - b. `DocumentReadAction`: impostato su `TEXTRACT_DETECT_DOCUMENT_TEXT`.
 - c. `FeatureTypes`— Non richiesto.

2. Usa l'AnalyzeDocumentAPI Amazon Textract per tutti i file PDF e di immagine.
 - a. DocumentReadMode: impostato su FORCE_DOCUMENT_READ_ACTION.
 - b. DocumentReadAction: impostato su TEXTTRACT_ANALYZE_DOCUMENT.
 - c. FeatureTypes— Impostato su TABLES FORMS o entrambe le funzionalità.
3. Usa l'AnalyzeDocumentAPI Amazon Textract per i file PDF scansionati e tutti i file di immagine.
 - a. DocumentReadMode: impostato su SERVICE_DEFAULT.
 - b. DocumentReadAction: impostato su TEXTTRACT_ANALYZE_DOCUMENT.
 - c. FeatureTypes— Impostato su FORMS o entrambe TABLES le funzionalità.

Per ulteriori informazioni sulle opzioni di Amazon Textract, consulta. [DocumentReaderConfig](#)

Procedure consigliate per le immagini

Quando utilizzi file di immagine per la classificazione personalizzata o il riconoscimento personalizzato delle entità, utilizza le seguenti linee guida per ottenere i migliori risultati:

- Fornite un'immagine di alta qualità, idealmente almeno 150 DPI.
- Se il file di immagine utilizza uno dei formati supportati (TIFF, JPEG o PNG), non convertire o eseguire il downsampling del file prima di caricarlo su Amazon S3.

Per ottenere risultati ottimali quando estrai il testo dalle tabelle nei documenti, segui queste pratiche:

- Le tabelle del documento sono visivamente separate dagli elementi circostanti della pagina. Ad esempio, la tabella non è sovrapposta a un'immagine o a un motivo complesso.
- Il testo all'interno della tabella è in posizione verticale. Ad esempio, il testo non viene ruotato rispetto al resto del testo della pagina.

Quando estraete il testo dalle tabelle, potreste vedere risultati incoerenti nei seguenti casi:

- Le celle di tabella unite si estendono su più colonne.
- Le tabelle hanno celle, righe o colonne diverse dalle altre parti della stessa tabella.

Classificazione personalizzata

Utilizza la classificazione personalizzata per organizzare i documenti in categorie (classi) da te definite. La classificazione personalizzata è un processo in due fasi. Innanzitutto, addestrate un modello di classificazione personalizzato (chiamato anche classificatore) per riconoscere le classi che vi interessano. Quindi utilizzate il modello per classificare un numero qualsiasi di set di documenti.

Ad esempio, è possibile classificare il contenuto delle richieste di supporto in modo da indirizzare la richiesta al team di supporto appropriato. Oppure puoi classificare le e-mail ricevute dai clienti per fornire indicazioni in base al tipo di richiesta del cliente. Puoi combinare Amazon Comprehend con Amazon Transcribe per convertire la voce in testo e quindi classificare le richieste provenienti dalle chiamate telefoniche di assistenza.

Puoi eseguire una classificazione personalizzata su un singolo documento in modo sincrono (in tempo reale) o avviare un processo asincrono per classificare un set di documenti. Puoi avere più classificatori personalizzati nel tuo account, ognuno addestrato utilizzando dati diversi. La classificazione personalizzata supporta una varietà di tipi di documenti di input, come testo semplice, PDF, Word e immagini.

Quando invii un lavoro di classificazione, scegli il modello di classificatore da utilizzare, in base al tipo di documenti che devi analizzare. Ad esempio, per analizzare documenti in testo semplice, è possibile ottenere risultati più accurati utilizzando un modello addestrato con documenti di testo semplice. Per analizzare documenti semistruutturati (come PDF, Word, immagini, output di Amazon Textract o file scansionati), ottieni i risultati più accurati utilizzando un modello che hai addestrato con documenti nativi.

Argomenti

- [Preparazione dei dati di addestramento del classificatore](#)
- [Modelli di classificazione della formazione](#)
- [Eseguire analisi in tempo reale](#)
- [Esecuzione di processi asincroni](#)

Preparazione dei dati di addestramento del classificatore

Per una classificazione personalizzata, si addestra il modello in modalità multiclasse o in modalità multietichetta. La modalità multiclasse associa una singola classe a ciascun documento. La modalità

multi-etichetta associa una o più classi a ciascun documento. I formati dei file di input sono diversi per ogni modalità, quindi scegliete la modalità da utilizzare prima di creare i dati di allenamento.

Note

La console Amazon Comprehend fa riferimento alla modalità multiclasse come modalità a etichetta singola.

La classificazione personalizzata supporta modelli addestrati con documenti di testo semplice e modelli addestrati con documenti nativi (come PDF, Word o immagini). Per ulteriori informazioni sui modelli di classificazione e sui tipi di documenti supportati, vedere. [Modelli di classificazione della formazione](#)

Per preparare i dati per addestrare un modello di classificatore personalizzato:

1. Identifica le classi che desideri che questo classificatore analizzi. Decidi quale modalità usare (multiclasse o multi-etichetta).
2. Decidi il tipo di modello di classificatore, in base al fatto che il modello sia destinato all'analisi di documenti di testo semplice o di documenti semistrutturati.
3. Raccogli esempi di documenti per ciascuna classe. Per i requisiti minimi di formazione, consulta [Quote generali per la classificazione dei documenti](#).
4. Per un modello in testo semplice, scegliete il formato del file di addestramento da utilizzare (file CSV o file manifesto aumentato). Per addestrare un modello di documento nativo, utilizzate sempre un file CSV.

Argomenti

- [Formati di file di formazione Classifier](#)
- [Modalità multiclasse](#)
- [Modalità multi-etichetta](#)

Formati di file di formazione Classifier

Per un modello in testo semplice, puoi fornire i dati di addestramento del classificatore come file CSV o come file manifest aumentato creato utilizzando AI Ground Truth. SageMaker Il file CSV o file manifesto aumentato include il testo di ogni documento di formazione e le etichette associate.

Per un modello di documento nativo, fornisci i dati di formazione di Classifier come file CSV. Il file CSV include il nome del file per ogni documento di formazione e le etichette associate. Includi i documenti di formazione nella cartella di input di Amazon S3 per il lavoro di formazione.

File CSV

Fornisci dati di allenamento etichettati come testo con codifica UTF-8 in un file CSV. Non includere una riga di intestazione. L'aggiunta di una riga di intestazione nel file può causare errori di runtime.

Per ogni riga del file CSV, la prima colonna contiene una o più etichette di classe. Un'etichetta di classe può essere qualsiasi stringa UTF-8 valida. Ti consigliamo di utilizzare nomi di classe chiari che non si sovrappongano nel significato. Il nome può includere spazi bianchi e può essere composto da più parole collegate da caratteri di sottolineatura o trattini.

Non lasciate spazi prima o dopo le virgole che separano i valori in una riga.

Il contenuto esatto del file CSV dipende dalla modalità di classificazione e dal tipo di dati di allenamento. Per i dettagli, consulta le sezioni relative a [Modalità multiclasse](#) e [Modalità multi-etichetta](#).

File manifesto aumentato

Un file manifest aumentato è un set di dati etichettato creato utilizzando AI Ground SageMaker Truth. Ground Truth è un servizio di etichettatura dei dati che aiuta te, o la forza lavoro che impieghi, a creare set di dati di formazione per modelli di apprendimento automatico.

Per ulteriori informazioni su Ground Truth e sull'output che produce, consulta [Use SageMaker AI Ground Truth to Label Data](#) nella Amazon SageMaker AI Developer Guide.

I file manifest aumentati sono in formato righe JSON. In questi file, ogni riga è un oggetto JSON completo che contiene un documento di formazione e le etichette associate. Il contenuto esatto di ogni riga dipende dalla modalità di classificazione. Per i dettagli, consulta le sezioni relative a [Modalità multiclasse](#) e [Modalità multi-etichetta](#).

Quando fornisci i dati di addestramento ad Amazon Comprehend, specifichi uno o più nomi di attributi dell'etichetta. Il numero di nomi di attributi che specifichi dipende dal fatto che il file manifesto aumentato sia l'output di un singolo processo di etichettatura o di un processo di etichettatura concatenato.

Se il file è l'output di un singolo lavoro di etichettatura, specifica il nome dell'attributo dell'etichetta singola dal lavoro Ground Truth.

Se il file è l'output di un lavoro di etichettatura concatenato, specifica il nome dell'attributo di etichetta per uno o più lavori della catena. Il nome di ogni attributo dell'etichetta fornisce le annotazioni di un singolo lavoro. È possibile specificare fino a 5 di questi attributi per i file manifest aumentati provenienti da processi di etichettatura concatenati.

Per ulteriori informazioni sui lavori di etichettatura concatenati e per esempi dei risultati che producono, consulta [Chaining Labeling Jobs nella](#) Amazon SageMaker AI Developer Guide.

Modalità multiclasse

In modalità multiclasse, la classificazione assegna una classe per ogni documento. Le singole classi si escludono a vicenda. Ad esempio, puoi classificare un film come commedia o fantascienza, ma non entrambi.

Note

La console Amazon Comprehend fa riferimento alla modalità multiclasse come modalità a etichetta singola.

Argomenti

- [Modelli in testo semplice](#)
- [Modelli di documenti nativi](#)

Modelli in testo semplice

Per addestrare un modello in testo semplice, puoi fornire dati di addestramento etichettati come file CSV o come file manifest aumentato da AI Ground Truth. SageMaker

File CSV

Per informazioni generali sull'utilizzo dei file CSV per i classificatori di formazione, consulta. [File CSV](#)

Fornisci i dati di formazione come file CSV a due colonne. Per ogni riga, la prima colonna contiene il valore dell'etichetta della classe. La seconda colonna contiene un documento di testo di esempio per quella classe. Ogni riga deve terminare con `\no\ r\ncaratteri`.

L'esempio seguente mostra un file CSV contenente tre documenti.

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

L'esempio seguente mostra una riga di un file CSV che addestra un classificatore personalizzato per rilevare se un messaggio di posta elettronica è spam:

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

File manifesto aumentato

Per informazioni generali sull'utilizzo dei file manifest aumentati per i classificatori di addestramento, vedere. [File manifesto aumentato](#)

Per i documenti di testo semplice, ogni riga del file manifest aumentato è un oggetto JSON completo che contiene un documento di formazione, un nome di classe singolo e altri metadati di Ground Truth. L'esempio seguente è un file manifest aumentato per addestrare un classificatore personalizzato a riconoscere i messaggi e-mail di 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"}}
```

L'esempio seguente mostra un oggetto JSON del file manifest aumentato, formattato per la leggibilità:

```
{
  "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"
}
```

In questo esempio, l'`source` attributo fornisce il testo del documento di formazione e assegna l'`MultiClassJob` indice di una classe da un elenco di classificazione. L'`job-name` attributo è il nome che hai definito per il lavoro di etichettatura in Ground Truth.

Quando inizi il processo di formazione sui classificatori in Amazon Comprehend, specifichi lo stesso nome del processo di etichettatura.

Modelli di documenti nativi

Un modello di documento nativo è un modello che si addestra con documenti nativi (come PDF, DOCX e immagini). I dati di addestramento vengono forniti come file CSV.

File CSV

Per informazioni generali sull'utilizzo dei file CSV per i classificatori di formazione, consulta [File CSV](#).

Fornisci i dati di formazione come file CSV a tre colonne. Per ogni riga, la prima colonna contiene il valore dell'etichetta della classe. La seconda colonna contiene il nome di file di un documento di esempio per questa classe. La terza colonna contiene il numero di pagina. Il numero di pagina è facoltativo se il documento di esempio è un'immagine.

L'esempio seguente mostra un file CSV che fa riferimento a tre documenti di input.

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

L'esempio seguente mostra una riga di un file CSV che addestra un classificatore personalizzato per rilevare se un messaggio di posta elettronica è spam. La pagina 2 del file PDF contiene l'esempio di spam.

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

Modalità multi-etichetta

In modalità multietichetta, le singole classi rappresentano categorie diverse che non si escludono a vicenda. La classificazione multietichetta assegna una o più classi a ciascun documento. Ad esempio, puoi classificare un film come documentario e un altro film come fantascienza, azione e commedia.

Per quanto riguarda la formazione, la modalità multi-etichetta supporta fino a 1 milione di esempi contenenti fino a 100 classi uniche.

Argomenti

- [Modelli in testo semplice](#)
- [Modelli di documenti nativi](#)

Modelli in testo semplice

Per addestrare un modello in testo semplice, puoi fornire dati di addestramento etichettati come file CSV o come file manifest aumentato da AI Ground Truth. SageMaker

File CSV

Per informazioni generali sull'utilizzo dei file CSV per i classificatori di formazione, consulta. [File CSV](#)

Fornisci i dati di formazione come file CSV a due colonne. Per ogni riga, la prima colonna contiene i valori delle etichette delle classi e la seconda contiene un documento di testo di esempio per queste classi. Per inserire più di una classe nella prima colonna, utilizzate un delimitatore (ad esempio un |) tra ogni classe.

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

L'esempio seguente mostra una riga di un file CSV che addestra un classificatore personalizzato per rilevare i generi negli abstract dei film:

```
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?"
```

Il delimitatore predefinito tra i nomi delle classi è una pipe (|). Tuttavia, è possibile utilizzare un carattere diverso come delimitatore. Il delimitatore deve essere distinto da tutti i caratteri dei nomi delle classi. Ad esempio, se le classi sono CLASS_1, CLASS_2 e CLASS_3, il carattere di sottolineatura (_) fa parte del nome della classe. Quindi non usate un carattere di sottolineatura come delimitatore per separare i nomi delle classi.

File manifesto aumentato

Per informazioni generali sull'utilizzo dei file manifest aumentati per i classificatori di addestramento, vedere [File manifesto aumentato](#)

Per i documenti in testo semplice, ogni riga del file manifest aumentato è un oggetto JSON completo. Contiene un documento di formazione, i nomi delle classi e altri metadati di Ground Truth. L'esempio seguente è un file manifest aumentato per addestrare un classificatore personalizzato a rilevare i generi negli abstract dei film:

```
{
  "source": "Document 1 text",
  "MultiLabelJob": [0,4],
  "MultiLabelJob-metadata": {
    "job-name": "labeling-job/multilabeljob",
    "class-map": {
      "0": "action",
      "4": "drama"
    },
    "human-annotated": "yes",
    "creation-date": "2020-05-21T19:02:21.521882",
    "confidence-map": {
      "0": 0.66
    },
    "type": "groundtruth/text-classification-multilabel"
  }
},
{
  "source": "Document 2 text",
  "MultiLabelJob": [3,6],
  "MultiLabelJob-metadata": {
    "job-name": "labeling-job/multilabeljob",
    "class-map": {
      "3": "comedy",
      "6": "horror"
    },
    "human-annotated": "yes",
    "creation-date": "2020-05-21T19:00:01.291202",
    "confidence-map": {
      "1": 0.61,
      "0": 0.61
    },
    "type": "groundtruth/text-classification-multilabel"
  }
},
{
  "source": "Document 3 text",
  "MultiLabelJob": [1],
  "MultiLabelJob-metadata": {
    "job-name": "labeling-job/multilabeljob",
    "class-map": {
      "1": "action"
    },
    "human-annotated": "yes",
    "creation-date": "2020-05-21T18:58:51.662050",
    "confidence-map": {
      "1": 0.68
    },
    "type": "groundtruth/text-classification-multilabel"
  }
}
```

L'esempio seguente mostra un oggetto JSON del file manifest aumentato, formattato per la leggibilità:

```
{
  "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"
    }
  }
}

```

In questo esempio, l'`source` attributo fornisce il testo del documento di formazione e assegna gli `MultiLabelJob` indici di diverse classi da un elenco di classificazione. Il nome del lavoro nei `MultiLabelJob` metadati è il nome che hai definito per il lavoro di etichettatura in Ground Truth.

Modelli di documenti nativi

Un modello di documento nativo è un modello che si addestra con documenti nativi (come PDF, DOCX e file di immagine). Fornisci dati di addestramento etichettati come file CSV.

File CSV

Per informazioni generali sull'utilizzo dei file CSV per i classificatori di formazione, consulta. [File CSV](#)

Fornisci i dati di formazione come file CSV a tre colonne. Per ogni riga, la prima colonna contiene i valori dell'etichetta della classe. La seconda colonna contiene il nome di file di un documento di esempio per queste classi. La terza colonna contiene il numero di pagina. Il numero di pagina è facoltativo se il documento di esempio è un'immagine.

Per inserire più di una classe nella prima colonna, utilizzate un delimitatore (ad esempio un `|`) tra ogni classe.

```
CLASS,input-doc-1.pdf,3
```

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

L'esempio seguente mostra una riga di un file CSV che addestra un classificatore personalizzato per rilevare i generi negli abstract dei film. La pagina 2 del file PDF contiene l'esempio di un film. comedy/teen

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

Il delimitatore predefinito tra i nomi delle classi è una pipe (|). Tuttavia, è possibile utilizzare un carattere diverso come delimitatore. Il delimitatore deve essere distinto da tutti i caratteri dei nomi delle classi. Ad esempio, se le classi sono CLASS_1, CLASS_2 e CLASS_3, il carattere di sottolineatura (_) fa parte del nome della classe. Quindi non usate un carattere di sottolineatura come delimitatore per separare i nomi delle classi.

Modelli di classificazione della formazione

Per addestrare un modello per la classificazione personalizzata, è necessario definire le categorie e fornire documenti di esempio per addestrare il modello personalizzato. Il modello viene addestrato in modalità multiclasse o multi-etichetta. La modalità multiclasse associa una singola classe a ciascun documento. La modalità multi-etichetta associa una o più classi a ciascun documento.

La classificazione personalizzata supporta due tipi di modelli di classificazione: modelli di testo semplice e modelli di documenti nativi. Un modello di testo semplice classifica i documenti in base al loro contenuto testuale. Un modello di documento nativo classifica anche i documenti in base al contenuto del testo. Un modello di documento nativo può anche utilizzare segnali aggiuntivi, ad esempio dal layout del documento. Si addestra un modello di documento nativo con documenti nativi affinché il modello apprenda le informazioni sul layout.

I modelli in testo semplice hanno le seguenti caratteristiche:

- Il modello viene addestrato utilizzando documenti di testo con codifica UTF-8.
- È possibile addestrare il modello utilizzando documenti in una delle seguenti lingue: inglese, spagnolo, tedesco, italiano, francese o portoghese.
- I documenti di formazione per un determinato classificatore devono utilizzare tutti la stessa lingua.
- I documenti di formazione sono in testo semplice, quindi non ci sono costi aggiuntivi per l'estrazione del testo.

I modelli di documenti nativi hanno le seguenti caratteristiche:

- Il modello viene addestrato utilizzando documenti semistrutturati, che includono i seguenti tipi di documenti:
 - Documenti PDF digitali e scansionati.
 - Documenti Word (DOCX).
 - Immagini: file JPG, file PNG e file TIFF a pagina singola.
 - L'API Textract genera file JSON.
- Il modello viene addestrato utilizzando documenti in inglese.
- Se i documenti di formazione includono file di documenti scansionati, dovrai sostenere costi aggiuntivi per l'estrazione del testo. Per ulteriori informazioni, consulta la pagina [dei prezzi di Amazon Comprehend](#).

Puoi classificare qualsiasi tipo di documento supportato utilizzando entrambi i tipi di modello. Tuttavia, per risultati più accurati, consigliamo di utilizzare un modello di testo semplice per classificare i documenti in testo semplice e un modello di documento nativo per classificare i documenti semistrutturati.

Argomenti

- [Addestra classificatori personalizzati \(console\)](#)
- [Addestra classificatori personalizzati \(API\)](#)
- [Verifica i dati di allenamento](#)
- [Risultati formativi di Classifier](#)
- [Metriche di classificazione personalizzate](#)

Addestra classificatori personalizzati (console)

È possibile creare e addestrare un classificatore personalizzato utilizzando la console e quindi utilizzare il classificatore personalizzato per analizzare i documenti.

Per addestrare un classificatore personalizzato, è necessario un set di documenti di formazione. Etichettate questi documenti con le categorie che desiderate che il classificatore di documenti riconosca. Per informazioni sulla preparazione dei documenti di formazione, consulta [Preparazione dei dati di addestramento del classificatore](#).

Per creare e addestrare un modello di classificazione dei documenti

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Personalizzazione, quindi scegli Classificazione personalizzata.
3. Scegli Crea nuovo modello.
4. In Impostazioni modello, inserisci un nome di modello per il classificatore. Il nome deve essere univoco all'interno del tuo account e della regione corrente.

(Facoltativo) Inserisci il nome della versione. Il nome deve essere univoco all'interno del tuo account e della regione corrente.

5. Seleziona la lingua dei documenti di formazione. Per vedere le lingue supportate dai classificatori, vedi. [Modelli di classificazione della formazione](#)
6. (Facoltativo) Se desideri crittografare i dati nel volume di storage mentre Amazon Comprehend elabora il tuo processo di formazione, scegli la crittografia Classifier. Quindi scegli se utilizzare una chiave KMS associata al tuo account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, scegli l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'ID della chiave in ARN della chiave KMS.

 Note

[Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, vedi \(\).AWS Key Management ServiceAWS KMS](#)

7. In Specifiche dei dati, scegli il tipo di modello di addestramento da utilizzare.
 - Documenti di testo semplice: scegliete questa opzione per creare un modello di testo semplice. Addestra il modello utilizzando documenti di testo semplice.
 - Documenti nativi: scegliete questa opzione per creare un modello di documento nativo. Addestra il modello utilizzando documenti nativi (PDF, Word, immagini).
8. Scegli il formato dei dati di allenamento. Per informazioni sui formati dei dati, vedere [Formati di file di formazione Classifier](#).

- File CSV: scegliete questa opzione se i dati di allenamento utilizzano il formato di file CSV.
 - Manifesto aumentato: scegli questa opzione se hai utilizzato Ground Truth per creare file manifest aumentati per i tuoi dati di allenamento. Questo formato è disponibile se avete scelto Documenti in testo semplice come tipo di modello di allenamento.
9. Scegliete la modalità Classifier da usare.
- Modalità etichetta singola: scegli questa modalità se le categorie che stai assegnando ai documenti si escludono a vicenda e stai addestrando il tuo classificatore ad assegnare un'etichetta a ciascun documento. Nell'API Amazon Comprehend, la modalità a etichetta singola è nota come modalità multiclasse.
 - Modalità multi-etichetta: scegli questa modalità se è possibile applicare più categorie a un documento contemporaneamente e stai addestrando il classificatore ad assegnare una o più etichette a ciascun documento.
10. Se scegli la modalità Multi-etichetta, puoi selezionare il Delimitatore per le etichette. Utilizzate questo carattere delimitatore per separare le etichette quando vi sono più classi per un documento di formazione. Il delimitatore predefinito è il carattere pipe.
11. (Facoltativo) Se avete scelto Augmented manifest come formato di dati, potete inserire fino a cinque file manifest aumentati. Ogni file manifesto aumentato contiene un set di dati di addestramento o un set di dati di test. È necessario fornire almeno un set di dati di addestramento. I set di dati di test sono facoltativi. Utilizzate i seguenti passaggi per configurare i file manifest aumentati:
- a. In Training and test dataset, espandi il pannello Input location.
 - b. In Tipo di set di dati, scegli Dati di addestramento o Dati di test.
 - c. Per la posizione S3 del file manifesto aumentato SageMaker AI Ground Truth, inserisci la posizione del bucket Amazon S3 che contiene il file manifest o accedi ad esso scegliendo Browse S3. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso per il processo di formazione deve avere autorizzazioni di lettura per il bucket S3.
 - d. Per i nomi degli attributi, inserisci il nome dell'attributo che contiene le tue annotazioni. Se il file contiene annotazioni provenienti da più lavori di etichettatura concatenati, aggiungi un attributo per ogni lavoro.
 - e. Per aggiungere un'altra posizione di input, scegliete Aggiungi posizione di input e quindi configurate la posizione successiva.

12. (Facoltativo) Se hai scelto il file CSV come formato di dati, utilizza i seguenti passaggi per configurare il set di dati di addestramento e il set di dati di test opzionale:
- In Training dataset, inserisci la posizione del bucket Amazon S3 che contiene il file CSV dei dati di allenamento o accedi al file selezionando Browse S3. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso al processo di formazione deve disporre delle autorizzazioni di lettura per il bucket S3.

(Facoltativo) Se hai scelto Documenti nativi come tipo di modello di formazione, fornisci anche l'URL della cartella Amazon S3 che contiene i file di esempio di formazione.
 - In Set di dati di test, seleziona se stai fornendo dati aggiuntivi ad Amazon Comprehend per testare il modello addestrato.
 - Autosplit: Autosplit seleziona automaticamente il 10% dei dati di allenamento da riservare per l'uso come dati di test.
 - (Facoltativo) Fornito dal cliente: inserisci l'URL del file CSV dei dati di test in Amazon S3. Puoi anche accedere alla sua posizione in Amazon S3 e scegliere Seleziona cartella.

(Facoltativo) Se scegli Documenti nativi come tipo di modello di formazione, fornisci anche l'URL della cartella Amazon S3 che contiene i file di test.
13. (Facoltativo) Per la modalità di lettura dei documenti, puoi sovrascrivere le azioni di estrazione del testo predefinite. Questa opzione non è richiesta per i modelli con testo semplice, poiché si applica all'estrazione del testo per i documenti scansionati. Per ulteriori informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).
14. (Facoltativo per i modelli in testo semplice) Per i dati di output, inserisci la posizione di un bucket Amazon S3 per salvare i dati di output dell'allenamento, come la matrice di confusione. Per ulteriori informazioni, consulta [Matrice di confusione](#).
- (Facoltativo) Se scegli di crittografare il risultato dell'output del tuo processo di formazione, scegli Crittografia. Quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
- Se utilizzi una chiave associata all'account corrente, scegli l'alias della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'alias o l'ID della chiave in KMS key ID.

15. Per il ruolo IAM, scegli un ruolo IAM esistente, quindi scegli un ruolo IAM esistente con autorizzazioni di lettura per il bucket S3 che contiene i tuoi documenti di formazione. Il ruolo deve avere una politica di fiducia che inizi con `comprehend.amazonaws.com` per essere valido.

Se non disponi già di un ruolo IAM con queste autorizzazioni, scegli Crea un ruolo IAM per crearne uno. Scegli le autorizzazioni di accesso per concedere questo ruolo, quindi scegli un suffisso di nome per distinguere il ruolo dai ruoli IAM nel tuo account.

 Note

Per i documenti di input crittografati, anche il ruolo IAM utilizzato deve disporre dell'autorizzazione `kms:Decrypt`. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per utilizzare la crittografia KMS](#).

16. (Facoltativo) Per avviare le tue risorse in Amazon Comprehend da un VPC, inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
 1. Scegli la sottorete in Sottoreti. Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.

 Note

Quando utilizzi un VPC con il tuo lavoro di classificazione, quello `DataAccessRole` utilizzato per le operazioni di creazione e avvio deve disporre delle autorizzazioni per il VPC che accede ai documenti di input e al bucket di output.

17. (Facoltativo) Per aggiungere un tag al classificatore personalizzato, inserite una coppia chiave-valore in Tag. Seleziona Aggiungi tag. Per rimuovere questa coppia prima di creare il classificatore, scegliete Rimuovi tag. Per ulteriori informazioni, consulta [Tagging delle risorse](#).
18. Scegli Create (Crea).

La console visualizza la pagina Classificatori. Il nuovo classificatore viene visualizzato nella tabella e ne viene visualizzato lo `Submitted` stato. Quando il classificatore inizia a elaborare i documenti di formazione, lo stato cambia in `Training`. Quando un classificatore è pronto per l'uso, lo stato

diventa o. Trained Trained with warnings Se lo stato è TRAINED_WITH_WARNINGS, esamina la cartella dei file ignorati in. [Risultati formativi di Classifier](#)

Se Amazon Comprehend ha riscontrato errori durante la creazione o la formazione, lo stato cambia in. In error Puoi scegliere un lavoro di classificatore nella tabella per ottenere maggiori informazioni sul classificatore, inclusi eventuali messaggi di errore.

Classifiers				
Stop Copy Delete Manage tags Create job Train classifier				
Q Search All < 1 > ⚙				
	Name	Training started	Training ended	Status
☐	classifiertags5-copy	7/22/2019, 3:48:38 PM	7/22/2019, 3:57:18 PM	✓ Trained
☐	classifiertags5	6/24/2019, 3:40:28 PM	6/24/2019, 3:47:26 PM	✓ Trained
☐	classifiertags!	6/3/2019, 6:33:16 PM	6/3/2019, 6:33:35 PM	✗ In error
☐	hk-classifier-output-2	4/9/2019, 11:28:26 AM	4/9/2019, 11:28:29 AM	✗ In error

Addestra classificatori personalizzati (API)

Per creare e addestrare un classificatore personalizzato, utilizzate l'[CreateDocumentClassifier](#) operazione.

È possibile monitorare l'avanzamento della richiesta utilizzando l'[DescribeDocumentClassifier](#) operazione. Dopo la transizione del Status campo a TRAINED, è possibile utilizzare il classificatore per classificare i documenti. Se lo stato è TRAINED_WITH_WARNINGS, esaminate la cartella dei file ignorati contenuta nel modulo dell'[Risultati formativi di Classifier](#) operazione. CreateDocumentClassifier

Argomenti

- [Addestramento: classificazione personalizzata utilizzando il AWS Command Line Interface](#)
- [Usare AWS SDK per Java o SDK per Python](#)

Addestramento: classificazione personalizzata utilizzando il AWS Command Line Interface

Negli esempi seguenti viene illustrato come utilizzare l'CreateDocumentClassifieroperazione, l'DescribeDocumentClassificationJoboperazione e altri classificatori personalizzati APIs con. AWS CLI

Gli esempi sono formattati per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Crea un classificatore personalizzato in testo semplice utilizzando l'operazione. 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
```

Per creare un classificatore personalizzato nativo, fornite i seguenti parametri aggiuntivi nella richiesta. create-document-classifier

1. DocumentType: impostate il valore su SEMI_STRUCTURED_DOCUMENT.
2. Documenti: la posizione S3 per i documenti di formazione (e, facoltativamente, i documenti dei test).
3. OutputDataConfig: fornisce la posizione S3 per i documenti di output (e una chiave KMS opzionale).
4. DocumentReaderConfig: Campo opzionale per le impostazioni di estrazione del testo.

```
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 \  
    S3Uri=s3://S3Bucket/docclass/file name \  
    DocumentType \  
      Documents
```

```
--data-access-role-arn arn:aws:iam::account number:role/testFlywheelDataAccess
```

Ottieni informazioni su un classificatore personalizzato con il classificatore di documenti ARN utilizzando l'operazione. `DescribeDocumentClassifier`

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

Eliminare un classificatore personalizzato utilizzando l'operazione. `DeleteDocumentClassifier`

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

Elenca tutti i classificatori personalizzati presenti nell'account utilizzando l'operazione.

`ListDocumentClassifiers`

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

Usare AWS SDK per Java o SDK per Python

Per esempi SDK su come creare e addestrare un classificatore personalizzato, consulta. [Utilizzo CreateDocumentClassifier con un AWS SDK o una CLI](#)

Verifica i dati di allenamento

Dopo aver addestrato il modello, Amazon Comprehend verifica il modello di classificatore personalizzato. Se non fornisci un set di dati di test, Amazon Comprehend addestra il modello con il 90% dei dati di addestramento. Riserva il 10 per cento dei dati di addestramento da utilizzare per i test. Se fornisci un set di dati di test, i dati del test devono includere almeno un esempio per ogni etichetta univoca nel set di dati di addestramento.

Il test del modello fornisce metriche che è possibile utilizzare per stimare l'accuratezza del modello. La console visualizza le metriche nella sezione delle prestazioni del classificatore della pagina dei dettagli del classificatore nella console. Vengono inoltre restituite nei `Metrics` campi restituiti dall'operazione. [DescribeDocumentClassifier](#)

Nel seguente esempio di dati di formazione, ci sono cinque etichette: DOCUMENTARY, DOCUMENTARY, SCIENCE_FICTION, DOCUMENTARY, ROMANTIC_COMEDY. Esistono tre classi uniche: DOCUMENTARY, SCIENCE_FICTION, ROMANTIC_COMEDIA.

Colonna 1	Colonna 2
DOCUMENTARIO	testo del documento 1
DOCUMENTARIO	testo del documento 2
FANTASCIENZA	testo del documento 3
DOCUMENTARIO	testo del documento 4
COMEDIA ROMANTICA	testo del documento 5

Per la suddivisione automatica (in cui Amazon Comprehend riserva il 10% dei dati di addestramento da utilizzare per i test), se i dati di addestramento contengono esempi limitati di un'etichetta specifica, il set di dati di test può contenere zero esempi di tale etichetta. Ad esempio, se il set di dati di addestramento contiene 1000 istanze della classe DOCUMENTARY, 900 istanze di SCIENCE_FICTION e una singola istanza della classe ROMANTIC_COMEDY, il set di dati di test potrebbe contenere 100 istanze DOCUMENTARY e 90 istanze SCIENCE_FICTION, ma nessuna istanza ROMANTIC_COMEDY, poiché è disponibile un solo esempio.

Dopo aver completato l'addestramento del modello, le metriche di addestramento forniscono informazioni che è possibile utilizzare per decidere se il modello è sufficientemente preciso per le proprie esigenze.

Risultati formativi di Classifier

Dopo aver completato la formazione sul modello di classificatore personalizzato, Amazon Comprehend crea file di output nella posizione di output di Amazon S3 specificata nella richiesta [CreateDocumentClassifier](#) API o nella richiesta di console equivalente.

Amazon Comprehend crea una matrice di confusione quando si addestra un modello di testo semplice o un modello di documento nativo. Può creare file di output aggiuntivi quando si addestra un modello di documento nativo.

Argomenti

- [Matrice di confusione](#)
- [Output aggiuntivi per modelli di documenti nativi](#)

Matrice di confusione

Quando si addestra un modello di classificatore personalizzato, Amazon Comprehend crea una matrice di confusione che fornisce metriche sulle prestazioni del modello durante la formazione. Questa matrice mostra una matrice di etichette prevista dal modello rispetto alle etichette effettive dei documenti. Amazon Comprehend utilizza una parte dei dati di addestramento per creare la matrice di confusione.

Una matrice di confusione fornisce un'indicazione di quali classi potrebbero utilizzare più dati per migliorare le prestazioni del modello. Una classe con un'alta percentuale di previsioni corrette ha il maggior numero di risultati lungo la diagonale della matrice. Se il numero sulla diagonale è un numero inferiore, la classe ha una frazione inferiore di previsioni corrette. Puoi aggiungere altri esempi di formazione per questa classe e addestrare nuovamente il modello. Ad esempio, se il 40 per cento dei campioni dell'etichetta A viene classificato come etichetta D, l'aggiunta di altri campioni per l'etichetta A e l'etichetta D migliora le prestazioni del classificatore.

Dopo che Amazon Comprehend ha creato il modello di classificazione, la matrice di confusione è disponibile nel `confusion_matrix.json` file nella posizione di output di S3.

Il formato della matrice di confusione varia a seconda che il classificatore sia stato addestrato utilizzando la modalità multiclasse o la modalità multi-etichetta.

Argomenti

- [Matrice di confusione per la modalità multiclasse](#)
- [Matrice di confusione per la modalità multietichetta](#)

Matrice di confusione per la modalità multiclasse

In modalità multiclasse, le singole classi si escludono a vicenda, pertanto la classificazione assegna un'etichetta a ciascun documento. Ad esempio, un animale può essere un cane o un gatto, ma non entrambi allo stesso tempo.

Considera il seguente esempio di matrice di confusione per un classificatore addestrato multiclasse:

```
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)

```

In questo caso, il modello prevedeva quanto segue:

- Un'etichetta «A» è stata prevista con precisione, due etichette «A» sono state erroneamente previste come etichette «B» e quattro etichette «A» sono state previste erroneamente come etichette «Y».
- Tre etichette «B» sono state previste con precisione e un'etichetta «B» è stata erroneamente prevista come etichetta «Y».
- Una «X» è stata prevista con precisione.
- Un'etichetta «Y» è stata prevista con precisione, una è stata prevista erroneamente come etichetta «A», una è stata prevista erroneamente come etichetta «B» e un'altra è stata prevista erroneamente come etichetta «X».

La linea diagonale nella matrice (A:A, B:B, X:X e Y:Y) mostra le previsioni accurate. Gli errori di previsione sono i valori esterni alla diagonale. In questo caso, la matrice mostra i seguenti tassi di errore di previsione:

- Etichette A: 86%
- Etichette B: 25%
- Etichette X: 0%
- Etichette Y: 75%

Il classificatore restituisce la matrice di confusione come file in formato JSON. Il seguente file JSON rappresenta la matrice dell'esempio precedente.

```

{
  "type": "multi_class",
  "confusion_matrix": [
    [1, 2, 0, 4],
    [0, 3, 0, 1],
    [0, 0, 1, 0],

```



```
[1, 1, 1, 1]],
"labels": ["A", "B", "X", "Y"],
"all_labels": ["A", "B", "X", "Y"]
}
```

Matrice di confusione per la modalità multietichetta

In modalità multietichetta, la classificazione può assegnare una o più classi a un documento. Considerate il seguente esempio di matrice di confusione per un classificatore addestrato a più classi.

In questo esempio, ci sono tre etichette possibili: ComedyAction, e. Drama La matrice di confusione multietichetta crea una matrice 2x2 per ogni etichetta.

Comedy			Action			Drama			<-(predicted label)
	No	Yes		No	Yes		No	Yes	
No	2	1	No	1	1	No	3	0	
Yes	0	2	Yes	2	1	Yes	1	1	
^			^			^			
------(was this label actually used)-----									

In questo caso, il modello ha restituito quanto segue per l'etichetta: Comedy

- Due casi in cui è stata prevista con precisione la presenza di Comedy un'etichetta. Vero positivo (TP).
- Due casi in cui è stata prevista con precisione l'assenza di Comedy un'etichetta. Vero negativo (TN).
- Zero casi in cui era stata erroneamente prevista la presenza di Comedy un'etichetta. Falso positivo (FP).
- Un caso in cui era stata erroneamente prevista l'assenza di un'Comedyetichetta. Falso negativo (FN).

Come in una matrice di confusione multiclasse, la linea diagonale di ogni matrice mostra le previsioni accurate.

In questo caso, il modello ha previsto con precisione Comedy le etichette l'80% delle volte (TP più TN) e le ha previste erroneamente il 20% delle volte (FP più FN).

Il classificatore restituisce la matrice di confusione come file in formato JSON. Il seguente file JSON rappresenta la matrice dell'esempio precedente.

```
{
  "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"]
}
```

Output aggiuntivi per modelli di documenti nativi

Amazon Comprehend può creare file di output aggiuntivi durante l'addestramento di un modello di documento nativo.

Output di Amazon Textract

Se Amazon Comprehend ha richiamato Amazon Textract per estrarre il testo APIs da uno qualsiasi dei documenti di formazione, salva i file di output di Amazon Textract nella posizione di output di S3. Utilizza la seguente struttura di directory:

- Documenti di formazione:

```
amazon-textract-output/train/<file_name>/<page_num>/textract_output.json
```

- Documenti di prova:

```
amazon-textract-output/test/<file_name>/<page_num>/textract_output.json
```

Amazon Comprehend compila la cartella di test se hai fornito documenti di test nella richiesta API.

Errori di annotazione dei documenti

Amazon Comprehend crea i seguenti file nella posizione di output di Amazon S3 (nella cartella `skipped_documents/`) in caso di annotazioni non riuscite:

- `failed_annotations_train.jsonl`

Il file esiste se qualche annotazione non è riuscita nei dati di addestramento.

- `failed_annotations_test.jsonl`

Il file esiste se la richiesta includeva dati di test e qualsiasi annotazione non è riuscita nei dati di test.

I file di annotazione non riusciti sono file JSONL con il seguente formato:

```
{
  "File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
{"File": "String", "Page": Number, "ErrorCode": "...", "ErrorMessage": "..."}
}
```

Metriche di classificazione personalizzate

Amazon Comprehend fornisce metriche per aiutarti a stimare le prestazioni di un classificatore personalizzato. Amazon Comprehend calcola le metriche utilizzando i dati di test del processo di formazione del classificatore. Le metriche rappresentano accuratamente le prestazioni del modello durante l'allenamento, quindi approssimano le prestazioni del modello per la classificazione di dati simili.

Utilizza operazioni API, ad esempio [DescribeDocumentClassifier](#) per recuperare le metriche per un classificatore personalizzato.

Note

Fai riferimento a [Metrics: Precision, recall e FScore](#) per una comprensione delle metriche sottostanti del punteggio Precision, Recall e F1. Queste metriche sono definite a livello di classe. Amazon Comprehend utilizza la media delle macro per combinare questi parametri nel set di test P, R e F1, come illustrato di seguito.

Argomenti

- [Metriche](#)
- [Miglioramento delle prestazioni del classificatore personalizzato](#)

Metriche

Amazon Comprehend supporta le seguenti metriche:

Argomenti

- [Accuratezza](#)
- [Precisione \(precisione macro\)](#)
- [Richiamo \(richiamo di macro\)](#)
- [Punteggio F1 \(punteggio macro F1\)](#)
- [Perdita di Hamming](#)
- [Micro precisione](#)
- [Micro richiamo](#)
- [Punteggio Micro F1](#)

Per visualizzare le metriche di un classificatore, apri la pagina dei dettagli del classificatore nella console.

Classifier performance Info			
Accuracy	Precision	Recall	F1 score
0.34	0.3298	0.3304	0.32
Hamming loss	Micro precision	Micro recall	Micro F1 score
-	-	-	-

Accuratezza

La precisione indica la percentuale di etichette dei dati di test che il modello ha previsto con precisione. Per calcolare l'accuratezza, dividi il numero di etichette previste con precisione nei documenti di test per il numero totale di etichette nei documenti di test.

Ad esempio

Etichetta effettiva	Etichetta prevista	Preciso/errato
1	1	Preciso
0	1	Errato
2	3	Errato
3	3	Preciso
2	2	Preciso
1	1	Preciso
3	3	Preciso

L'accuratezza è costituita dal numero di previsioni accurate diviso per il numero di campioni complessivi dei test = $5/7 = 0,714$ o 71,4%

Precisione (precisione macro)

La precisione è una misura dell'utilità dei risultati del classificatore nei dati del test. È definito come il numero di documenti classificati accuratamente, diviso per il numero totale di classificazioni per la classe. L'alta precisione significa che il classificatore ha restituito risultati significativamente più pertinenti rispetto a quelli non pertinenti.

La *Precision* metrica è anche nota come Macro Precision.

L'esempio seguente mostra i risultati di precisione per un set di test.

Etichetta	Dimensione del campione	Precisione dell'etichetta
Etichetta_1	400	0.75
Etichetta_2	300	0,80
Etichetta_3	30000	0.90
Etichetta_4	20	0.50

Etichetta	Dimensione del campione	Precisione dell'etichetta
Etichetta_5	10	0,40

La metrica Precision (Macro Precision) per il modello è quindi:

$$\text{Macro Precision} = (0.75 + 0.80 + 0.90 + 0.50 + 0.40)/5 = 0.67$$

Richiamo (richiamo di macro)

Indica la percentuale di categorie corrette nel testo che il modello è in grado di prevedere. Questa metrica deriva dalla media dei punteggi di richiamo di tutte le etichette disponibili. Il richiamo è una misura della completezza dei risultati del classificatore per i dati del test.

Un richiamo elevato significa che il classificatore ha restituito la maggior parte dei risultati pertinenti.

La Recall metrica è anche nota come Macro Recall.

L'esempio seguente mostra i risultati di richiamo per un set di test.

Etichetta	Dimensione del campione	Richiamo dell'etichetta
Etichetta_1	400	0,70
Etichetta_2	300	0,70
Etichetta_3	30000	0,98
Etichetta_4	20	0,80
Etichetta_5	10	0.10

La metrica Recall (Macro Recall) per il modello è quindi:

$$\text{Macro Recall} = (0.70 + 0.70 + 0.98 + 0.80 + 0.10)/5 = 0.656$$

Punteggio F1 (punteggio macro F1)

Il punteggio F1 è derivato dai valori and. Precision Recall Misura la precisione complessiva del classificatore. Il punteggio più alto è 1 e il punteggio più basso è 0.

Amazon Comprehend calcola il punteggio Macro F1. È la media non ponderata dei punteggi F1 dell'etichetta. Utilizzando il seguente set di test come esempio:

Etichetta	Dimensione del campione	Etichetta il punteggio F1
Etichetta_1	400	0,724
Etichetta_2	300	0,824
Etichetta_3	30000	0,94
Etichetta_4	20	0,62
Etichetta_5	10	0,16

Il punteggio F1 (Macro F1 Score) per il modello viene calcolato come segue:

$$\text{Macro F1 Score} = (0.724 + 0.824 + 0.94 + 0.62 + 0.16)/5 = 0.6536$$

Perdita di Hamming

La frazione di etichette prevista in modo errato. Considerata anche come la frazione di etichette errate rispetto al numero totale di etichette. I punteggi più vicini allo zero sono migliori.

Micro precisione

Originale:

Simile alla metrica di precisione, tranne per il fatto che la microprecisione si basa sul punteggio complessivo di tutti i punteggi di precisione sommati.

Micro richiamo

Simile alla metrica di richiamo, tranne per il fatto che il micro richiamo si basa sul punteggio complessivo di tutti i punteggi di richiamo sommati.

Punteggio Micro F1

Il punteggio Micro F1 è una combinazione delle metriche Micro Precision e Micro Recall.

Miglioramento delle prestazioni del classificatore personalizzato

Le metriche forniscono una panoramica delle prestazioni del classificatore personalizzato durante un processo di classificazione. Se le metriche sono basse, il modello di classificazione potrebbe non essere efficace per il tuo caso d'uso. Sono disponibili diverse opzioni per migliorare le prestazioni del classificatore:

1. Nei dati di allenamento, fornisci esempi concreti che definiscano una netta separazione delle categorie. Ad esempio, fornite documenti che utilizzano Unique words/sentences per rappresentare la categoria.
2. Aggiungi altri dati per le etichette sottorappresentate nei dati di allenamento.
3. Cerca di ridurre l'inclinazione nelle categorie. Se l'etichetta più grande dei tuoi dati contiene più di 10 volte i documenti nell'etichetta più piccola, prova ad aumentare il numero di documenti per l'etichetta più piccola. Assicurati di ridurre il rapporto di inclinazione a un massimo di 10:1 tra le classi altamente rappresentate e quelle meno rappresentate. Puoi anche provare a rimuovere i documenti di input dalle classi altamente rappresentate.

Eseguire analisi in tempo reale

Dopo aver addestrato un classificatore personalizzato, puoi classificare i documenti utilizzando l'analisi in tempo reale. L'analisi in tempo reale utilizza un singolo documento come input e restituisce i risultati in modo sincrono. La classificazione personalizzata accetta una varietà di tipi di documenti come input per l'analisi in tempo reale. Per informazioni dettagliate, vedi [Ingressi per analisi personalizzate in tempo reale](#).

Se prevedi di analizzare file di immagini o documenti PDF scansionati, la tua policy IAM deve concedere le autorizzazioni per utilizzare due metodi DetectDocumentText API Amazon Textract (e). AnalyzeDocument Amazon Comprehend richiama questi metodi durante l'estrazione del testo. Per un esempio di policy, consulta [Autorizzazioni necessarie per eseguire azioni di analisi dei documenti](#).

È necessario creare un endpoint per eseguire analisi in tempo reale utilizzando un modello di classificazione personalizzato.

Argomenti

- [Analisi in tempo reale per una classificazione personalizzata \(console\)](#)
- [Analisi in tempo reale per la classificazione personalizzata \(API\)](#)
- [Output per analisi in tempo reale](#)

Analisi in tempo reale per una classificazione personalizzata (console)

Puoi utilizzare la console Amazon Comprehend per eseguire analisi in tempo reale utilizzando un modello di classificazione personalizzato.

Crei un endpoint per eseguire l'analisi in tempo reale. Un endpoint include risorse gestite che rendono disponibile il modello personalizzato per l'inferenza in tempo reale.

Per informazioni sul provisioning della velocità effettiva degli endpoint e sui costi associati, consulta. [Utilizzo degli endpoint Amazon Comprehend](#)

Argomenti

- [Creazione di un endpoint per la classificazione personalizzata](#)
- [Esecuzione della classificazione personalizzata in tempo reale](#)

Creazione di un endpoint per la classificazione personalizzata

Per creare un endpoint (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Endpoints e scegli il pulsante Crea endpoint. Si apre una schermata Crea endpoint.
3. Assegna un nome all'endpoint. Il nome deve essere univoco all'interno della regione e dell'account correnti.
4. Scegli un modello personalizzato a cui desideri collegare il nuovo endpoint. Dal menu a discesa, puoi cercare per nome del modello.

Note

È necessario creare un modello prima di potervi collegare un endpoint. Se non hai ancora un modello, vedi [Modelli di classificazione della formazione](#).

5. (Facoltativo) per aggiungere un tag all'endpoint, inserisci una coppia chiave-valore in Tag e scegli Aggiungi tag. Per rimuovere questa coppia prima di creare l'endpoint, scegli Rimuovi tag
6. Immettete il numero di unità di inferenza (IUs) da assegnare all'endpoint. Ogni unità rappresenta una velocità effettiva di 100 caratteri al secondo per un massimo di due documenti al secondo. Per informazioni sulla velocità effettiva degli endpoint, vedere. [Utilizzo degli endpoint Amazon Comprehend](#)
7. (Facoltativo) Se state creando un nuovo endpoint, avete la possibilità di utilizzare lo stimatore dell'interfaccia utente. A seconda della velocità effettiva o del numero di caratteri da analizzare al secondo, può essere difficile sapere quante unità di inferenza sono necessarie. Questo passaggio facoltativo può aiutarti a determinare il numero IUs di richieste.
8. Nel riepilogo degli acquisti, esamina il costo finale orario, giornaliero e mensile stimato.
9. Seleziona la casella di controllo se ritieni che il tuo account comporti addebiti per l'endpoint dal momento in cui viene avviato fino a quando non lo elimini.
10. Scegli Crea endpoint

Esecuzione della classificazione personalizzata in tempo reale

Dopo aver creato un endpoint, puoi eseguire analisi in tempo reale utilizzando il tuo modello personalizzato. Esistono due modi per eseguire analisi in tempo reale dalla console. È possibile inserire testo o caricare un file, come illustrato di seguito.

Per eseguire analisi in tempo reale utilizzando un modello personalizzato (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analisi in tempo reale.
3. In Tipo di input, scegli Personalizzato per il tipo di analisi.
4. In Tipo di modello personalizzato, scegli Classificazione personalizzata.
5. Per Endpoint, scegli l'endpoint che desideri utilizzare. Questo endpoint si collega a un modello personalizzato specifico.
6. Per specificare i dati di input per l'analisi, puoi inserire del testo o caricare un file.
 - Per inserire del testo:
 - a. Scegli Inserisci testo.
 - b. Inserisci il testo che desideri analizzare.

- Per caricare un file:
 - a. Scegli Carica file e inserisci il nome del file da caricare.
 - b. (Facoltativo) In Azioni di lettura avanzate, puoi sostituire le azioni predefinite per l'estrazione del testo. Per maggiori dettagli, consultare [Impostazione delle opzioni di estrazione del testo](#).

Per ottenere risultati ottimali, abbinate il tipo di input al tipo di modello di classificatore. La console visualizza un avviso se inviate un documento nativo a un modello di testo normale o testo semplice a un modello di documento nativo. Per ulteriori informazioni, consulta [Modelli di classificazione della formazione](#).

7. Scegliete Analizza. Amazon Comprehend analizza i dati di input utilizzando il tuo modello personalizzato. Amazon Comprehend mostra le classi scoperte, insieme a una valutazione di confidenza per ogni classe.

Analisi in tempo reale per la classificazione personalizzata (API)

Puoi utilizzare l'API Amazon Comprehend per eseguire la classificazione in tempo reale con un modello personalizzato. Innanzitutto, crei un endpoint per eseguire l'analisi in tempo reale. Dopo aver creato l'endpoint, si esegue la classificazione in tempo reale.

Gli esempi in questa sezione utilizzano i formati di comandi per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Per informazioni sul provisioning della velocità effettiva degli endpoint e sui costi associati, consulta [Utilizzo degli endpoint Amazon Comprehend](#)

Argomenti

- [Creazione di un endpoint per la classificazione personalizzata](#)
- [Esecuzione della classificazione personalizzata in tempo reale](#)

Creazione di un endpoint per la classificazione personalizzata

L'esempio seguente mostra il funzionamento dell'[CreateEndpoint](#) API utilizzando AWS CLI

```
aws comprehend create-endpoint \
```

```
--desired-inference-units number of inference units \  
--endpoint-name endpoint name \  
--model-arn arn:aws:comprehend:region:account-id:model/example \  
--tags Key=My1stTag,Value=Value1
```

Amazon Comprehend risponde con quanto segue:

```
{  
  "EndpointArn": "Arn"  
}
```

Esecuzione della classificazione personalizzata in tempo reale

Dopo aver creato un endpoint per il modello di classificazione personalizzato, si utilizza l'endpoint per eseguire l'operazione [ClassifyDocument](#) API. È possibile fornire un input di testo utilizzando il parametro `text` o `bytes`. Immettete gli altri tipi di input utilizzando il `bytes` parametro.

Per i file di immagine e i file PDF, è possibile utilizzare il `DocumentReaderConfig` parametro per sovrascrivere le azioni di estrazione del testo predefinite. Per maggiori dettagli, consultare [Impostazione delle opzioni di estrazione del testo](#).

Per ottenere risultati ottimali, abbinate il tipo di input al tipo di modello di classificatore. La risposta dell'API include un avviso se invii un documento nativo a un modello di testo semplice o un file di testo semplice a un modello di documento nativo. Per ulteriori informazioni, consulta [Modelli di classificazione della formazione](#).

Usando il AWS Command Line Interface

Gli esempi seguenti mostrano come utilizzare il comando CLI `classify-document`.

Classificare il testo utilizzando il AWS CLI

L'esempio seguente esegue la classificazione in tempo reale su un blocco di testo.

```
aws comprehend classify-document \  
  --endpoint-arn arn:aws:comprehend:region:account-id:endpoint/endpoint name \  
  --text 'From the Tuesday, April 16th, 1912 edition of The Guardian newspaper: The  
maiden voyage of the White Star liner Titanic,  
the largest ship ever launched ended in disaster. The Titanic started her trip  
from Southampton for New York on Wednesday. Late
```

```
on Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress,
and several liners were near enough to catch and respond to the call.'
```

Amazon Comprehend risponde con quanto segue:

```
{
  "Classes": [
    {
      "Name": "string",
      "Score": 0.9793661236763
    }
  ]
}
```

Classificate un documento semistrutturato utilizzando il AWS CLI

Per analizzare la classificazione personalizzata di un file PDF, Word o di immagine, esegui il `classify-document` comando con il file di input nel `bytes` parametro.

L'esempio seguente utilizza un'immagine come file di input. Utilizza l'`fileb` opzione per codificare in base 64 i byte del file di immagine. Per ulteriori informazioni, vedere [Binary large objects](#) nella Guida per l'utente. AWS Command Line Interface

Questo esempio passa anche un file JSON denominato `config.json` per impostare le opzioni di estrazione del testo.

```
$ aws comprehend classify-document \
> --endpoint-arn arn \
> --language-code en \
> --bytes fileb://image1.jpg \
> --document-reader-config file://config.json
```

Il file `config.json` contiene il seguente contenuto.

```
{
  "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",
  "DocumentReadAction": "TEXTTRACT_DETECT_DOCUMENT_TEXT"
}
```

Amazon Comprehend risponde con quanto segue:

```
{
  "Classes": [
    {
      "Name": "string",
      "Score": 0.9793661236763
    }
  ]
}
```

Per ulteriori informazioni, consulta [ClassifyDocument](#) Amazon Comprehend API Reference.

Output per analisi in tempo reale

Uscite per input di testo

Per gli input di testo, l'output include l'elenco di classi o etichette identificate dall'analisi del classificatore. L'esempio seguente mostra un elenco con due classi.

```
"Classes": [
  {
    "Name": "abc",
    "Score": 0.2757999897003174,
    "Page": 1
  },
  {
    "Name": "xyz",
    "Score": 0.2721000015735626,
    "Page": 1
  }
]
```

Uscite per input semistruutturati

Per un documento di input semistruutturato o un file di testo, l'output può includere i seguenti campi aggiuntivi:

- **DocumentMetadata** — Informazioni di estrazione sul documento. I metadati includono un elenco di pagine del documento, con il numero di caratteri estratti da ciascuna pagina. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **DocumentType** — Il tipo di documento per ogni pagina del documento di input. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **Errori**: errori a livello di pagina rilevati dal sistema durante l'elaborazione del documento di input. Il campo è vuoto se il sistema non ha riscontrato errori.
- **Avvertenze**: avvisi rilevati durante l'elaborazione del documento di input. La risposta include un avviso in caso di mancata corrispondenza tra il tipo di documento di input e il tipo di modello associato all'endpoint specificato. Il campo è vuoto se il sistema non ha generato alcun avviso.

Per ulteriori dettagli su questi campi di output, consulta [ClassifyDocument](#) Amazon Comprehend API Reference.

L'esempio seguente mostra l'output di un documento di input PDF nativo di una pagina.

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

Esecuzione di processi asincroni

Dopo aver addestrato un classificatore personalizzato, è possibile utilizzare processi asincroni per analizzare documenti di grandi dimensioni o più documenti in un unico batch.

La classificazione personalizzata accetta una varietà di tipi di documenti di input. Per informazioni dettagliate, vedi [Ingressi per analisi asincrone personalizzate](#).

Se prevedi di analizzare file di immagini o documenti PDF scansionati, la tua policy IAM deve concedere le autorizzazioni per utilizzare due metodi DetectDocumentText API Amazon Textract (e). AnalyzeDocument Amazon Comprehend richiama questi metodi durante l'estrazione del testo. Per un esempio di policy, consulta [Autorizzazioni necessarie per eseguire azioni di analisi dei documenti](#).

Per la classificazione di documenti semistruutturati (immagini, file PDF o Docx) utilizzando un modello di testo semplice, utilizza il formato di input. `one document per file` Inoltre, includi il parametro `DocumentReaderConfig` nella tua richiesta. [StartDocumentClassificationJob](#)

Argomenti

- [Formati di file per l'analisi asincrona](#)
- [Lavori di analisi per la classificazione personalizzata \(console\)](#)
- [Lavori di analisi per la classificazione personalizzata \(API\)](#)
- [Output per lavori di analisi asincroni](#)

Formati di file per l'analisi asincrona

Quando si esegue un'analisi asincrona con il modello, è possibile scegliere tra diversi formati per i documenti di input: `One document per line` o `one document per file` Il formato utilizzato dipende dal tipo di documenti che si desidera analizzare, come descritto nella tabella seguente.

Description	Formato
L'input contiene più file. Ogni file contiene un documento di input. Questo formato è ideale per raccolte di documenti di grandi dimensioni, come articoli di giornale o articoli scientifici. Inoltre, utilizzate questo formato per documenti semistruzzurati (file di immagine, PDF o Docx) utilizzando un classificatore di documenti nativo.	Un documento per file
L'input è costituito da uno o più file. Ogni riga del file è un documento di input separato. Questo formato è ideale per documenti brevi, come messaggi di testo o post sui social media.	Un documento per riga

Un documento per file

Con il `one document per file` formato, ogni file rappresenta un documento di input.

Un documento per riga

Con il `One document per line` formato, ogni documento viene posizionato su una riga separata e non viene utilizzata alcuna intestazione. L'etichetta non è inclusa in ogni riga (poiché non conosci ancora l'etichetta del documento). Ogni riga del file (la fine del singolo documento) deve terminare con un'alimentazione di riga (LF,\n), un corriage return (CR,\r) o entrambi (CRLF,\r\n). Non utilizzare il separatore di riga UTF-8 (u+2028) per terminare una riga.

L'esempio seguente mostra il formato del file di input.

```
Text of document 1 \n
Text of document 2 \n
Text of document 3 \n
Text of document 4 \n
```

Per entrambi i formati, utilizzate la codifica UTF-8 per i file di testo. Dopo aver preparato i file, inseriscili nel bucket S3 che stai utilizzando per i dati di input.

Quando inizi un processo di classificazione, specifichi questa posizione Amazon S3 per i dati di input. L'URI deve trovarsi nella stessa regione dell'endpoint API che stai chiamando. L'URI può puntare a un singolo file (come quando si utilizza il metodo «un documento per riga») oppure può essere il prefisso per una raccolta di file di dati.

Ad esempio, se utilizzi l'URI `S3://bucketName/prefix`, se il prefisso è un singolo file, Amazon Comprehend utilizza quel file come input. Se più di un file inizia con il prefisso, Amazon Comprehend li utilizza tutti come input.

Concedi ad Amazon Comprehend l'accesso al bucket S3 che contiene la raccolta di documenti e i file di output. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Lavori di analisi per la classificazione personalizzata (console)

Dopo aver creato e addestrato un [classificatore di documenti personalizzato](#), è possibile utilizzare la console per eseguire processi di classificazione personalizzati con il modello.

Per creare un processo di classificazione personalizzato (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analysis jobs, quindi scegli Crea job.
3. Assegna un nome al lavoro di classificazione. Il nome deve essere univoco per il tuo account e per la regione corrente.
4. In Tipo di analisi, scegli Classificazione personalizzata.
5. Da Seleziona classificatore, scegli il classificatore personalizzato da utilizzare.
6. (Facoltativo) Se scegli di crittografare i dati utilizzati da Amazon Comprehend durante l'elaborazione del lavoro, scegli Job encryption. Quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, scegli l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'ID della chiave in ARN della chiave KMS.

 Note

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta [Servizio di gestione delle chiavi](#) (KMS).

7. In Dati di input, inserisci la posizione del bucket Amazon S3 che contiene i documenti di input o accedi al bucket scegliendo Browse S3. Questo bucket deve trovarsi nella stessa regione dell'API che stai chiamando. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso per il processo di classificazione deve disporre delle autorizzazioni di lettura per il bucket S3.

Per ottenere il massimo livello di precisione nell'addestramento di un modello, abbina il tipo di input al tipo di modello di classificatore. Il processo di classificazione restituisce un avviso se si inviano documenti nativi a un modello di testo semplice o documenti di testo semplice a un modello di documento nativo. Per ulteriori informazioni, consulta [Modelli di classificazione della formazione](#).

8. (Facoltativo) Per il formato di input, è possibile scegliere il formato dei documenti di input. Il formato può essere un documento per file o un documento per riga in un singolo file. Un documento per riga si applica solo ai documenti di testo.
9. (Facoltativo) Per la modalità di lettura dei documenti, è possibile sovrascrivere le azioni di estrazione del testo predefinite. Per ulteriori informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).
10. In Dati di output, inserisci la posizione del bucket Amazon S3 in cui Amazon Comprehend deve scrivere i dati di output del lavoro o accedervi selezionando Browse S3. Questo bucket deve trovarsi nella stessa regione dell'API che stai chiamando. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso per il processo di classificazione deve avere autorizzazioni di scrittura per il bucket S3.
11. (Facoltativo) Se scegli di crittografare il risultato dell'output del tuo lavoro, scegli Crittografia. Quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, scegli l'alias o l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'alias o l'ID della chiave in KMS key ID.

12. (Facoltativo) Per avviare le tue risorse in Amazon Comprehend da un VPC, inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
 1. Scegli la sottorete in Subnet (s). Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.

Note

Quando utilizzi un VPC con il tuo lavoro di classificazione, quello `DataAccessRole` utilizzato per le operazioni di creazione e avvio deve concedere le autorizzazioni al VPC che accede al bucket di output.

13. Scegli Crea lavoro per creare il processo di classificazione dei documenti.

Lavori di analisi per la classificazione personalizzata (API)

Dopo aver [creato e addestrato](#) un classificatore di documenti personalizzato, è possibile utilizzare il classificatore per eseguire lavori di analisi.

Utilizzate l'[StartDocumentClassificationJob](#) operazione per iniziare a classificare i documenti senza etichetta. Si specifica il bucket S3 che contiene i documenti di input, il bucket S3 per i documenti di output e il classificatore da utilizzare.

Per ottenere il massimo livello di precisione nell'addestramento di un modello, abbina il tipo di input al tipo di modello di classificatore. Il processo di classificazione restituisce un avviso se si inviano documenti nativi a un modello di testo semplice o documenti di testo semplice a un modello di documento nativo. Per ulteriori informazioni, consulta [Modelli di classificazione della formazione](#).

[StartDocumentClassificationJob](#) è asincrono. Una volta avviato il processo, utilizzate l'[DescribeDocumentClassificationJob](#) operazione per monitorarne l'avanzamento. Quando il Status campo della risposta viene visualizzato COMPLETED, è possibile accedere all'output nella posizione specificata.

Argomenti

- [Utilizzando il AWS Command Line Interface](#)

- [Usare AWS SDK per Java o SDK per Python](#)

Utilizzando il AWS Command Line Interface

I seguenti esempi sono l'StartDocumentClassificationJoboperazione e altri classificatori personalizzati APIs con. AWS CLI

Gli esempi seguenti utilizzano il formato dei comandi per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Esegui un processo di classificazione personalizzato utilizzando l'StartDocumentClassificationJoboperazione.

```
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
```

Otteni informazioni su un classificatore personalizzato con l'ID del lavoro utilizzando l'DescribeDocumentClassificationJoboperazione.

```
aws comprehend describe-document-classification-job \  
  --region region \  
  --job-id job id
```

Elenca tutti i lavori di classificazione personalizzati presenti nel tuo account utilizzando l'ListDocumentClassificationJobsoperazione.

```
aws comprehend list-document-classification-jobs  
  --region region
```

Usare AWS SDK per Java o SDK per Python

Per esempi SDK su come avviare un processo di classificazione personalizzato, consulta. [Utilizzo StartDocumentClassificationJob con un AWS SDK o una CLI](#)

Output per lavori di analisi asincroni

Al termine di un processo di analisi, memorizza i risultati nel bucket S3 specificato nella richiesta.

Uscite per input di testo

Per entrambi i formati di documenti di input di testo (multiclasse o multietichetta), l'output del lavoro è costituito da un singolo file denominato `output.tar.gz`. È un file di archivio compresso che contiene un file di testo con l'output.

Output multiclasse

Quando si utilizza un classificatore addestrato in modalità multiclasse, i risultati vengono visualizzati. `classes` Ognuna di queste `classes` è la classe utilizzata per creare l'insieme di categorie durante l'addestramento del classificatore.

Per ulteriori dettagli su questi campi di output, consulta [ClassifyDocument](#) Amazon Comprehend API Reference.

Gli esempi seguenti utilizzano le seguenti classi che si escludono a vicenda.

```
DOCUMENTARY
SCIENCE_FICTION
ROMANTIC_COMEDY
SERIOUS_DRAMA
OTHER
```

Se il formato dei dati di input è un documento per riga, il file di output contiene una riga per ogni riga dell'input. Ogni riga include il nome del file, il numero di riga a base zero della riga di input e la classe o le classi presenti nel documento. Termina con la certezza che Amazon Comprehend ha che la singola istanza sia stata classificata correttamente.

Esempio:

```
{"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 il formato dei dati di input è un documento per file, il file di output contiene una riga per ogni documento. Ogni riga contiene il nome del file e la classe o le classi presenti nel documento. Termina con la certezza che Amazon Comprehend abbia classificato accuratamente la singola istanza.

Esempio:

```
{"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": "Documentary", "Score": 0.0372}]}
{"File": "file3.txt", "Classes": [{"Name": "Serious_Drama", "Score": 0.3141}, {"Name": "Other", "Score": 0.0381}, {"Name": "Other", "Score": 0.0372}]}
```

Output multietichetta

Quando si utilizza un classificatore addestrato in modalità multietichetta, i risultati vengono visualizzati. labels Ognuna di queste labels è l'etichetta utilizzata per creare l'insieme di categorie durante l'addestramento del classificatore.

Gli esempi seguenti utilizzano queste etichette uniche.

```
SCIENCE_FICTION
ACTION
DRAMA
COMEDY
ROMANCE
```

Se il formato dei dati di input è un documento per riga, il file di output contiene una riga per ogni riga dell'input. Ogni riga include il nome del file, il numero di riga a base zero della riga di input e la classe o le classi presenti nel documento. Termina con la certezza che Amazon Comprehend ha che la singola istanza sia stata classificata correttamente.

Esempio:

```
{"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 il formato dei dati di input è un documento per file, il file di output contiene una riga per ogni documento. Ogni riga contiene il nome del file e la classe o le classi presenti nel documento. Termina con la certezza che Amazon Comprehend abbia classificato accuratamente la singola istanza.

Esempio:

```
{
  "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 }
  ]
}
```

Output per documenti di input semistrutturati

Per i documenti di input semistrutturati, l'output può includere i seguenti campi aggiuntivi:

- **DocumentMetadata** — Informazioni di estrazione sul documento. I metadati includono un elenco di pagine del documento, con il numero di caratteri estratti da ciascuna pagina. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **DocumentType** — Il tipo di documento per ogni pagina del documento di input. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **Errori**: errori a livello di pagina rilevati dal sistema durante l'elaborazione del documento di input. Il campo è vuoto se il sistema non ha riscontrato errori.

Per ulteriori dettagli su questi campi di output, consulta [ClassifyDocument](#) Amazon Comprehend API Reference.

L'esempio seguente mostra l'output di un file PDF scansionato di due pagine.


```
[{ #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"
}]
```

Riconoscimento personalizzato delle entità

Il riconoscimento personalizzato delle entità estende le funzionalità di Amazon Comprehend aiutandoti a identificare nuovi tipi di entità specifici che non rientrano nei tipi di entità [generici](#) preimpostati. Ciò significa che puoi analizzare documenti ed estrarre entità come codici di prodotto o entità specifiche dell'azienda che soddisfano le tue esigenze particolari.

La creazione autonoma di un riconoscimento di entità accurato e personalizzato può essere un processo complesso, che richiede la preparazione di grandi set di documenti di formazione annotati manualmente e la selezione degli algoritmi e dei parametri corretti per l'addestramento dei modelli. Amazon Comprehend aiuta a ridurre la complessità fornendo annotazioni automatiche e lo sviluppo di modelli per creare un modello di riconoscimento delle entità personalizzato.

La creazione di un modello di riconoscimento delle entità personalizzato è un approccio più efficace rispetto all'utilizzo di corrispondenze di stringhe o espressioni regolari per estrarre entità dai documenti. Ad esempio, per estrarre i nomi ENGINEER in un documento, è difficile enumerare tutti i nomi possibili. Inoltre, senza contesto, è difficile distinguere tra i nomi ENGINEER e i nomi ANALYST. Un modello di riconoscimento delle entità personalizzato può apprendere il contesto in cui è probabile che compaiano tali nomi. Inoltre, la corrispondenza tra stringhe non rileva le entità che presentano errori di battitura o seguono nuove convenzioni di denominazione, mentre ciò è possibile utilizzando un modello personalizzato.

Sono disponibili due opzioni per creare un modello personalizzato:

1. Annotazioni: forniscono un set di dati contenente entità annotate per l'addestramento del modello.
2. Elenchi di entità (solo testo semplice): forniscono un elenco di entità e la relativa etichetta di tipo (ad esempio, un set di documenti non annotati contenenti tali entità per l'addestramento dei modelli). `PRODUCT_CODES`

Quando si crea un riconoscimento di entità personalizzato utilizzando file PDF annotati, è possibile utilizzarlo con una varietà di formati di file di input: testo semplice, file di immagine (JPG, PNG, TIFF), file PDF e documenti Word, senza necessità di preelaborazione o appiattimento del documento. Amazon Comprehend non supporta l'annotazione di file di immagine o documenti Word.

 Note

Un riconoscitore di entità personalizzato che utilizza file PDF annotati supporta solo documenti in inglese.

È possibile addestrare un modello su un massimo di 25 entità personalizzate contemporaneamente. Per maggiori dettagli, consulta la [pagina Linee guida e quote](#).

Dopo aver addestrato il modello, puoi utilizzarlo per il rilevamento delle entità in tempo reale e nei lavori di rilevamento delle entità.

Argomenti

- [Preparazione dei dati di formazione di Entity Recognizer](#)
- [Addestramento di modelli di riconoscimento di entità personalizzati](#)
- [Esecuzione di analisi di riconoscimento personalizzate in tempo reale](#)
- [Esecuzione di processi di analisi per il riconoscimento personalizzato delle entità](#)

Preparazione dei dati di formazione di Entity Recognizer

Per addestrare un modello di riconoscimento delle entità personalizzato di successo, è importante fornire al model trainer dati di alta qualità come input. Senza dati validi, il modello non imparerà a identificare correttamente le entità.

Puoi scegliere uno dei due modi per fornire dati ad Amazon Comprehend per addestrare un modello di riconoscimento delle entità personalizzato:

- **Elenco di entità:** elenca le entità specifiche in modo che Amazon Comprehend possa addestrarsi a identificare le entità personalizzate. Nota: gli elenchi di entità possono essere utilizzati solo per documenti di testo semplice.
- **Annotazioni:** fornisce la posizione delle entità in una serie di documenti in modo che Amazon Comprehend possa addestrarsi sia sull'entità che sul suo contesto. Per creare un modello per l'analisi di file di immagini o documenti Word PDFs, devi addestrare il tuo riconoscitore a utilizzare le annotazioni PDF.

In entrambi i casi, Amazon Comprehend apprende il tipo di documenti e il contesto in cui si trovano le entità e crea un sistema di riconoscimento che può generalizzare per rilevare le nuove entità quando analizzi i documenti.

Quando crei un modello personalizzato (o addestra una nuova versione), puoi fornire un set di dati di test. Se non fornisci dati di test, Amazon Comprehend riserva il 10% dei documenti di input per testare il modello. Amazon Comprehend addestra il modello con i documenti rimanenti.

Se fornisci un set di dati di test per il tuo set di formazione sulle annotazioni, i dati di test devono includere almeno un'annotazione per ciascuno dei tipi di entità specificati nella richiesta di creazione.

Argomenti

- [Quando utilizzare le annotazioni rispetto agli elenchi di entità](#)
- [Elenchi di entità \(solo testo semplice\)](#)
- [Annotazioni](#)

Quando utilizzare le annotazioni rispetto agli elenchi di entità

La creazione di annotazioni richiede più lavoro rispetto alla creazione di un elenco di entità, ma il modello risultante può essere molto più preciso. L'utilizzo di un elenco di entità è più rapido e meno impegnativo, ma i risultati sono meno raffinati e meno accurati. Questo perché le annotazioni forniscono più contesto per Amazon Comprehend da utilizzare durante l'addestramento del modello. Senza tale contesto, Amazon Comprehend avrà un numero maggiore di falsi positivi quando cercherà di identificare le entità.

Ci sono scenari in cui è più conveniente dal punto di vista commerciale evitare le spese e il carico di lavoro più elevati derivanti dall'uso delle annotazioni. Ad esempio, il nome John Johnson è importante per la ricerca, ma il fatto che si tratti della persona esatta non è rilevante. Oppure, le metriche utilizzate nell'elenco delle entità sono sufficientemente valide da fornirti i risultati di riconoscimento di cui hai bisogno. In questi casi, utilizzare invece un elenco di entità può essere la scelta più efficace.

Si consiglia di utilizzare la modalità annotazioni nei seguenti casi:

- Se prevedi di eseguire inferenze per file di immagine o documenti Word. PDFs In questo scenario, si addestra un modello utilizzando file PDF con annotazioni e si utilizza il modello per eseguire lavori di inferenza per file di immagine e documenti PDFs Word.
- Quando il significato delle entità potrebbe essere ambiguo e dipendente dal contesto. Ad esempio, il termine Amazon potrebbe riferirsi al fiume in Brasile o al rivenditore online Amazon.com. Quando

crei un identificatore di entità personalizzato per identificare entità aziendali come Amazon, dovresti utilizzare le annotazioni anziché un elenco di entità perché questo metodo consente di utilizzare meglio il contesto per trovare le entità.

- Quando ti senti a tuo agio nella configurazione di un processo per l'acquisizione delle annotazioni, l'operazione può richiedere un certo impegno.

Si consiglia di utilizzare un elenco di entità nei seguenti casi:

- Quando si dispone già di un elenco di entità o quando è relativamente semplice comporre un elenco completo di entità. Se utilizzi un elenco di entità, l'elenco deve essere completo o almeno includere la maggior parte delle entità valide che potrebbero apparire nei documenti che fornisci per la formazione.
- Per gli utenti alle prime armi, in genere si consiglia di utilizzare un elenco di entità perché ciò richiede uno sforzo minore rispetto alla creazione di annotazioni. Tuttavia, è importante notare che il modello addestrato potrebbe non essere accurato come se si utilizzassero le annotazioni.

Elenchi di entità (solo testo semplice)

Per addestrare un modello utilizzando un elenco di entità, si forniscono due informazioni: un elenco dei nomi delle entità con i corrispondenti tipi di entità personalizzati e una raccolta di documenti non annotati in cui si prevede che le entità appaiano.

Quando fornisci un elenco di entità, Amazon Comprehend utilizza un algoritmo intelligente per rilevare le occorrenze dell'entità nei documenti e fungere da base per l'addestramento del modello di riconoscimento delle entità personalizzato.

Per gli elenchi di entità, fornisci almeno 25 corrispondenze per tipo di entità nell'elenco delle entità.

Un elenco di entità per il riconoscimento personalizzato delle entità richiede un file con valori separati da virgole (CSV), con le seguenti colonne:

- Testo: il testo di un esempio di voce esattamente come visualizzato nel corpus del documento di accompagnamento.
- Tipo: il tipo di entità definito dal cliente. I tipi di entità devono contenere una stringa maiuscola separata da un carattere di sottolineatura, ad esempio MANAGER o SENIOR_MANAGER. È possibile addestrare fino a 25 tipi di entità per modello.

Il file `documents.txt` contiene quattro righe:

```
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.
```

Il file CSV con l'elenco delle entità contiene le seguenti righe:

```
Text, Type  
Jo Brown, ENGINEER  
John Doe, ENGINEER  
Jane Smith, MANAGER
```

 Note

Nell'elenco delle entità, la voce relativa a Emilio Johnson non è presente perché non contiene né l'entità ENGINEER né l'entità MANAGER.

Creazione dei file di dati

È importante che l'elenco delle entità sia contenuto in un file CSV configurato correttamente, quindi la possibilità di avere problemi con il file dell'elenco delle entità è minima. Per configurare manualmente il file CSV, deve essere vero quanto segue:

- La codifica UTF-8 deve essere specificata in modo esplicito, anche se viene utilizzata come impostazione predefinita nella maggior parte dei casi.
- Deve includere i nomi delle colonne: e. `Type Text`

Consigliamo vivamente di generare i file di input CSV a livello di codice per evitare potenziali problemi.

L'esempio seguente usa Python per generare un CSV per le annotazioni mostrate sopra:

```
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"])
```

Best practice

Esistono diversi fattori da considerare per ottenere il miglior risultato quando si utilizza un elenco di entità, tra cui:

- L'ordine delle entità nell'elenco non ha effetti sull'addestramento dei modelli.
- Utilizza gli elementi dell'elenco delle entità che coprono l'80%-100% degli esempi di entità positive menzionati nel corpus di documenti non annotati.
- Evita gli esempi di entità che corrispondono a entità diverse dal corpus del documento rimuovendo parole e frasi comuni. Anche una manciata di corrispondenze errate può influire in modo significativo sulla precisione del modello risultante. Ad esempio, una parola come quella presente nell'elenco delle entità produrrà un numero elevato di corrispondenze, che è improbabile che si tratti delle entità che state cercando, e quindi influirà in modo significativo sulla precisione.
- I dati di input non devono contenere duplicati. La presenza di campioni duplicati potrebbe causare la contaminazione del set di test e quindi influire negativamente sul processo di addestramento, sulle metriche del modello e sul comportamento.
- Fornisci documenti che assomiglino il più possibile a casi d'uso reali. Non utilizzare dati relativi a giocattoli o dati sintetizzati per i sistemi di produzione. I dati di input devono essere il più diversi possibile per evitare un adattamento eccessivo e aiutare il modello sottostante a generalizzare meglio su esempi reali.
- L'elenco delle entità fa distinzione tra maiuscole e minuscole e le espressioni regolari non sono attualmente supportate. Tuttavia, il modello addestrato è spesso in grado di riconoscere le entità anche se non corrispondono esattamente all'involucro fornito nell'elenco delle entità.
- Se hai un'entità che è una sottostringa di un'altra entità (come «Smith» e «Jane Smith»), forniscile entrambe nell'elenco delle entità.

Ulteriori suggerimenti sono disponibili all'indirizzo [Miglioramento delle prestazioni del riconoscimento di entità personalizzato](#)

Annotazioni

Le annotazioni etichettano le entità nel contesto associando i tipi di entità personalizzati alle posizioni in cui si trovano nei documenti di formazione.

Inviando le annotazioni insieme ai documenti, è possibile aumentare la precisione del modello. Con Annotations, non stai semplicemente fornendo la posizione dell'entità che stai cercando, ma stai anche fornendo un contesto più accurato all'entità personalizzata che stai cercando.

Ad esempio, se stai cercando il nome John Johnson, con il tipo di entità JUDGE, fornire la tua annotazione potrebbe aiutare il modello a capire che la persona che vuoi trovare è un giudice. Se è in grado di utilizzare il contesto, Amazon Comprehend non troverà persone di nome John Johnson che siano avvocati o testimoni. Senza fornire annotazioni, Amazon Comprehend creerà la propria versione di un'annotazione, ma non sarà altrettanto efficace nell'includere solo i giudici. Fornire annotazioni personalizzate può aiutare a ottenere risultati migliori e a generare modelli in grado di sfruttare meglio il contesto durante l'estrazione di entità personalizzate.

Argomenti

- [Numero minimo di annotazioni](#)
- [Le migliori pratiche di annotazione](#)
- [File di annotazione in testo semplice](#)
- [File di annotazioni PDF](#)
- [Annotazione di file PDF](#)

Numero minimo di annotazioni

Il numero minimo di documenti e annotazioni di input necessari per addestrare un modello dipende dal tipo di annotazioni.

Annotazioni PDF

Per creare un modello per l'analisi di file di immagini o documenti Word PDFs, addestrate il vostro riconoscitore utilizzando le annotazioni PDF. Per le annotazioni PDF, fornisci almeno 250 documenti di input e almeno 100 annotazioni per entità.

Se fornisci un set di dati di test, i dati di test devono includere almeno un'annotazione per ciascuno dei tipi di entità specificati nella richiesta di creazione.

Annotazioni in testo semplice

Per creare un modello per l'analisi dei documenti di testo, potete addestrare il vostro sistema di riconoscimento utilizzando annotazioni in testo semplice.

Per le annotazioni in testo semplice, fornite almeno tre documenti di input annotati e almeno 25 annotazioni per entità. Se fornischi meno di 50 annotazioni in totale, Amazon Comprehend riserva più del 10% dei documenti di input per testare il modello (a meno che tu non abbia fornito un set di dati di test nella richiesta di formazione). Non dimenticare che la dimensione minima del corpus di documenti è di 5 KB.

Se l'input contiene solo pochi documenti di formazione, è possibile che si verifichi un errore nel senso che i dati di input della formazione contengono troppo pochi documenti che menzionano una delle entità. Invia nuovamente il lavoro con documenti aggiuntivi che menzionano l'entità.

Se fornisci un set di dati di test, i dati del test devono includere almeno un'annotazione per ciascuno dei tipi di entità specificati nella richiesta di creazione.

Per un esempio di come eseguire il benchmark di un modello con un set di dati di piccole dimensioni, consulta [Amazon Comprehend annuncia limiti di annotazione inferiori per il riconoscimento personalizzato delle entità](#) sul sito del blog. AWS

Le migliori pratiche di annotazione

Esistono diversi fattori da considerare per ottenere risultati ottimali quando si utilizzano le annotazioni, tra cui:

- Annota i tuoi dati con cura e verifica di annotare ogni menzione dell'entità. Le annotazioni imprecise possono portare a risultati scadenti.
- I dati di input non devono contenere duplicati, come un duplicato di un PDF che intendi annotare. La presenza di un campione duplicato potrebbe causare la contaminazione del set di test e influire negativamente sul processo di addestramento, sulle metriche del modello e sul comportamento del modello.
- Assicurati che tutti i tuoi documenti siano annotati e che i documenti senza annotazioni siano dovuti alla mancanza di entità legittime, non a negligenza. Ad esempio, se hai un documento che dice «J Doe è ingegnere da 14 anni», dovresti fornire anche un'annotazione per «J Doe» e «John Doe». In caso contrario, il modello potrebbe confondere il modello e potrebbe non riconoscere «J Doe» come INGEGNERE. Ciò deve essere coerente all'interno dello stesso documento e tra i documenti.
- In generale, più annotazioni portano a risultati migliori.
- È possibile addestrare un modello con il [numero minimo](#) di documenti e annotazioni, ma l'aggiunta di dati di solito migliora il modello. Si consiglia di aumentare il volume dei dati annotati del 10% per aumentare la precisione del modello. È possibile eseguire l'inferenza su un set di dati di test

che rimane invariato e può essere testato con diverse versioni del modello. È quindi possibile confrontare le metriche per le versioni successive del modello.

- Fornisci documenti che assomiglino il più possibile a casi d'uso reali. I dati sintetizzati con schemi ripetitivi dovrebbero essere evitati. I dati di input dovrebbero essere il più diversi possibile per evitare un adattamento eccessivo e aiutare il modello sottostante a generalizzare meglio su esempi reali.
- È importante che i documenti siano diversi in termini di numero di parole. Ad esempio, se tutti i documenti contenuti nei dati di addestramento sono brevi, il modello risultante potrebbe avere difficoltà a prevedere le entità nei documenti più lunghi.
- Provate a fornire per l'addestramento la stessa distribuzione dei dati che vi aspettate di utilizzare quando state effettivamente rilevando le vostre entità personalizzate (tempo di inferenza). Ad esempio, al momento dell'inferenza, se prevedi di inviarti documenti che non contengono entità, anche questi dovrebbero far parte del tuo set di documenti di formazione.

Per ulteriori suggerimenti, consulta [Miglioramento delle prestazioni del riconoscimento di entità personalizzato](#).

File di annotazione in testo semplice

Per le annotazioni in testo semplice, si crea un file con valori separati da virgole (CSV) che contiene un elenco di annotazioni. Il file CSV deve contenere le seguenti colonne se il formato di input del file di training è un documento per riga.

File	Line (Linea)	Offset iniziale	Offset finale	Tipo
Il nome del file contenente il documento. Ad esempio, se uno dei file del documento si trova in <code>ins3://my-S3-bucket/test-files/document</code>	Il numero di riga contenente l'entità. Ometti questa colonna se il formato di input è un documento per file.	L'offset dei caratteri nel testo di input (rispetto all'inizio o della riga) che mostra dove inizia l'entità. Il primo carattere si trova nella posizione 0.	L'offset dei caratteri nel testo di input che mostra dove finisce l'entità.	Il tipo di entità definito dal cliente. I tipi di entità devono essere una stringa maiuscola separata da un carattere di sottolineatura. Consigliata.

File	Line (Linea)	Offset iniziale	Offset finale	Tipo
ts.txt , il valore nella File colonna sarà documents .txt . È necessari o includere l'estensione del file (in questo caso '.txt') come parte del nome del file.				mo di utilizzar e tipi di entità descrittivi come, o. MANAGER SENIOR_MA NAGER PRODUCT_C ODE È possibile addestrare fino a 25 tipi di entità per modello.

Se il formato di input del file di addestramento è un documento per file, si omette la colonna del numero di riga e i valori Offset iniziale e Offset finale sono gli offset dell'entità dall'inizio del documento.

L'esempio seguente riguarda un documento per riga. Il file documents.txt contiene quattro righe (righe 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.
```

Il file CSV con l'elenco delle annotazioni è il seguente:

```
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

Nel file di annotazioni, il numero di riga contenente l'entità inizia con la riga 0. In questo esempio, il file CSV non contiene alcuna voce per la riga 2 perché non c'è alcuna entità nella riga 2 di `documents.txt`

Creazione dei file di dati

È importante inserire le annotazioni in un file CSV configurato correttamente per ridurre il rischio di errori. Per configurare manualmente il file CSV, deve essere vero quanto segue:

- La codifica UTF-8 deve essere specificata in modo esplicito, anche se viene utilizzata come impostazione predefinita nella maggior parte dei casi.
- La prima riga contiene le intestazioni delle colonne: `File, Line (opzionale), Begin Offset End Offset Type`

Ti consigliamo vivamente di generare i file di input CSV a livello di codice per evitare potenziali problemi.

L'esempio seguente usa Python per generare un CSV per le annotazioni mostrate in precedenza:

```
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"])
```

File di annotazioni PDF

Per le annotazioni PDF, usi SageMaker AI Ground Truth per creare un set di dati etichettato in un file manifest aumentato. Ground Truth è un servizio di etichettatura dei dati che aiuta te (o la forza lavoro che impieghi) a creare set di dati di formazione per modelli di apprendimento automatico. Amazon Comprehend accetta file manifest aumentati come dati di addestramento per modelli personalizzati. Puoi fornire questi file quando crei un riconoscimento di entità personalizzato utilizzando la console Amazon Comprehend o [CreateEntityRecognizer](#) l'azione API.

È possibile utilizzare il tipo di attività integrato di Ground Truth, Named Entity Recognition, per creare un processo di etichettatura per consentire ai lavoratori di identificare le entità nel testo. Per ulteriori informazioni, consulta [Named Entity Recognition](#) nella Amazon SageMaker AI Developer Guide. Per ulteriori informazioni su Amazon SageMaker Ground Truth, consulta [Use Amazon SageMaker AI Ground Truth per etichettare i dati](#).

Note

Usando Ground Truth, puoi definire etichette sovrapposte (testo che associ a più di un'etichetta). Tuttavia, il riconoscimento delle entità Amazon Comprehend non supporta etichette sovrapposte.

I file manifest aumentati sono in formato righe JSON. In questi file, ogni riga è un oggetto JSON completo che contiene un documento di formazione e le etichette associate. L'esempio seguente è un file manifest aumentato che addestra un identificatore di entità a rilevare le professioni delle persone menzionate nel testo:

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

Ogni riga di questo file JSON lines è un oggetto JSON completo, in cui gli attributi includono il testo del documento, le annotazioni e altri metadati di Ground Truth. L'esempio seguente è un singolo oggetto JSON nel file manifest aumentato, ma è formattato per garantire la leggibilità:

```
{
  "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"
  }
}
```

In questo esempio, l'`source` attributo fornisce il testo del documento di formazione e l'`NamedEntityRecognitionDemo` attributo fornisce le annotazioni per le entità nel testo. Il nome dell'`NamedEntityRecognitionDemo` attributo è arbitrario e fornisci un nome a tua scelta quando definisci il lavoro di etichettatura in Ground Truth.

In questo esempio, l'`NamedEntityRecognitionDemo` attributo è il nome dell'attributo `label`, che è l'attributo che fornisce le etichette che un operatore di Ground Truth assegna ai dati di

addestramento. Quando fornisci i tuoi dati di addestramento ad Amazon Comprehend, devi specificare uno o più nomi di attributi dell'etichetta. Il numero di nomi di attributi che specifichi dipende dal fatto che il file manifesto aumentato sia l'output di un singolo processo di etichettatura o di un processo di etichettatura concatenato.

Se il file è l'output di un singolo processo di etichettatura, specifica il nome dell'attributo dell'etichetta singola che è stato utilizzato quando il lavoro è stato creato in Ground Truth.

Se il file è l'output di un lavoro di etichettatura concatenato, specifica il nome dell'attributo dell'etichetta per uno o più lavori della catena. Il nome di ogni attributo dell'etichetta fornisce le annotazioni di un singolo lavoro. È possibile specificare fino a 5 di questi attributi per i file manifest aumentati prodotti da processi di etichettatura concatenati.

In un file manifesto aumentato, il nome dell'attributo label segue in genere la chiave. `source` Se il file è l'output di un lavoro concatenato, ci saranno più nomi di attributi di etichetta. Quando fornisci i tuoi dati di addestramento ad Amazon Comprehend, fornisci solo gli attributi che contengono annotazioni pertinenti per il tuo modello. Non specificare gli attributi che terminano con «-metadata».

Per ulteriori informazioni sui lavori di etichettatura concatenati e per esempi dei risultati che producono, consulta [Chaining Labeling Jobs nella](#) Amazon SageMaker AI Developer Guide.

Annotazione di file PDF

Prima di poter annotare il tuo allenamento PDFs in SageMaker AI Ground Truth, completa i seguenti prerequisiti:

- Installa python3.8.x
- Installazione di [jq](#)
- Installa la [AWS CLI](#)

Se stai usando la regione us-east-1, puoi saltare l'installazione della CLI perché è già installata con AWS il tuo ambiente Python. In questo caso, crei un ambiente virtuale per usare Python 3.8 in Cloud9. AWS

- [AWS Configura](#) le tue credenziali
- Crea una [forza lavoro SageMaker AI Ground Truth](#) privata per supportare le annotazioni

Assicurati di registrare il nome del team di lavoro che hai scelto nella tua nuova forza lavoro privata, mentre lo usi durante l'installazione.

Argomenti

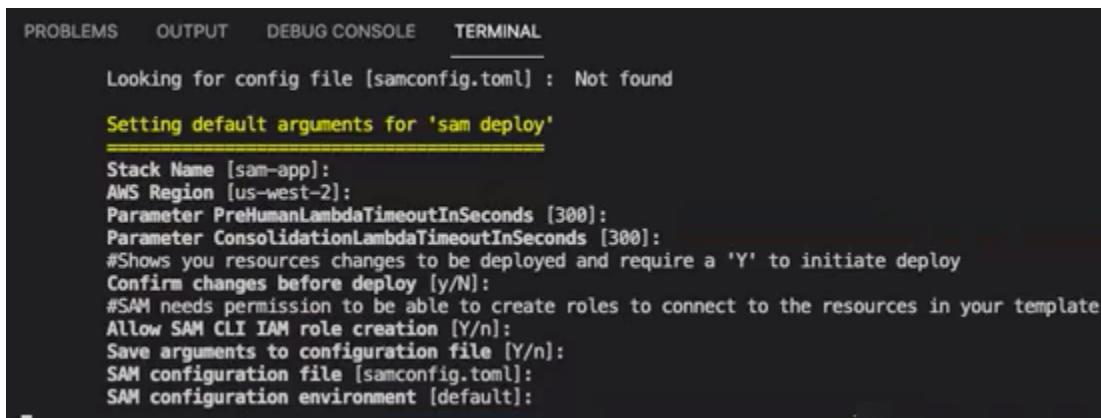
- [Configurazione dell'ambiente](#)
- [Caricamento di un PDF in un bucket S3](#)
- [Creazione di un lavoro di annotazione](#)
- [Annotazione con SageMaker AI Ground Truth](#)

Configurazione dell'ambiente

1. Se usi Windows, installa [Cygwin](#); se usi Linux o Mac, salta questo passaggio.
2. Scarica gli artefatti di [annotazione](#) da GitHub. Decomprimere il file.
3. Dalla finestra del terminale, accedete alla cartella decompressa (-). `amazon-comprehend-semi-structured documents-annotation-tools-main`
4. Questa cartella include una scelta da eseguire per installare le dipendenze, configurare un virtualenv Python e distribuire le risorse richieste. Makefiles Controlla il file readme per fare la tua scelta.
5. L'opzione consigliata utilizza un solo comando per installare tutte le dipendenze in un virtualenv, crea lo stack a partire dal modello e distribuisce lo CloudFormation stack all'utente con una guida interattiva. Account AWS Esegui il comando seguente:

```
make ready-and-deploy-guided
```

Questo comando presenta una serie di opzioni di configurazione. Assicurati che il tuo Regione AWS sia corretto. Per tutti gli altri campi, puoi accettare i valori predefiniti o inserire valori personalizzati. Se modifichi il nome CloudFormation dello stack, scrivilo quando ne hai bisogno nei passaggi successivi.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Looking for config file [samconfig.toml] : Not found

Setting default arguments for 'sam deploy'

Stack Name [sam-app]:
AWS Region [us-west-2]:
Parameter PreHumanLambdaTimeoutInSeconds [300]:
Parameter ConsolidationLambdaTimeoutInSeconds [300]:
#Shows you resources changes to be deployed and require a 'Y' to initiate deploy
Confirm changes before deploy [y/N]:
#SAM needs permission to be able to create roles to connect to the resources in your template
Allow SAM CLI IAM role creation [Y/n]:
Save arguments to configuration file [Y/n]:
SAM configuration file [samconfig.toml]:
SAM configuration environment [default]:
```


Lo CloudFormation stack crea e gestisce i [AWS lambda](#), i ruoli [AWS IAM](#) e i bucket [AWS S3](#) necessari per lo strumento di annotazione.

Puoi esaminare ciascuna di queste risorse nella pagina dei dettagli dello stack nella console CloudFormation

6. Il comando richiede di avviare la distribuzione. CloudFormation crea tutte le risorse nella regione specificata.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Deploying with following values
Stack name      : sam-app
Region          : us-west-2
Confirm changeset : False
Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-jnw8g1gm4pqh
Capabilities     : ["CAPABILITY_IAM"]
Parameter overrides : {"PreHumanLambdaTimeoutInSeconds": "300", "ConsolidationLambdaTimeoutInSeconds": "300"}
Signing Profiles : {}

Initiating deployment
```

Quando lo stato dello CloudFormation stack passa a creazione/completamento, le risorse sono pronte per l'uso.

Caricamento di un PDF in un bucket S3

Nella sezione [Configurazione](#), hai distribuito uno CloudFormation stack che crea un bucket S3 denominato `comprehend-semi-structured-documents AWS::Region}-${AWS::AccountId}`. Ora carichi i tuoi documenti PDF di origine in questo bucket.

Note

Questo secchio contiene i dati necessari per il tuo lavoro di etichettatura. La policy Lambda Execution Role concede l'autorizzazione alla funzione Lambda per accedere a questo bucket. Puoi trovare il nome del bucket S3 nei dettagli dello CloudFormation stack usando la chiave 'S3Bucket'. SemiStructuredDocuments

1. Crea una nuova cartella nel bucket S3. Assegna un nome a questa nuova cartella «src».
2. Aggiungi i tuoi file sorgente PDF alla cartella 'src'. In una fase successiva, aggiungi annotazioni a questi file per addestrare il tuo riconoscitore.

3. (Facoltativo) Ecco un esempio di AWS CLI che puoi usare per caricare i tuoi documenti sorgente da una directory locale in un bucket S3:

```
aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided/src/
```

Oppure, con la tua regione e il tuo ID account:

```
aws s3 cp --recursive local-path-to-your-source-docs s3://deploy-guided-Region-AccountID/src/
```

4. Ora hai una forza lavoro SageMaker AI Ground Truth privata e hai caricato i tuoi file sorgente nel bucket S3, `deploy-guided/src/`; sei pronto per iniziare ad annotare.

Creazione di un lavoro di annotazione

Lo script `comprehend-ssie-annotation-tool-cli.py` nella `bin` directory è un semplice comando wrapper che semplifica la creazione di un processo di etichettatura AI Ground SageMaker Truth. Lo script python legge i documenti sorgente dal bucket S3 e crea un corrispondente file manifest a pagina singola con un documento sorgente per riga. Lo script crea quindi un processo di etichettatura, che richiede il file manifest come input.

[Lo script python utilizza il bucket e lo CloudFormation stack S3 configurati nella sezione Configurazione.](#) I parametri di input richiesti per lo script includono:

- `input-s3-path`: Uri S3 per i documenti sorgente che hai caricato nel tuo bucket S3. Ad esempio: `s3://deploy-guided/src/`. Puoi anche aggiungere la tua regione e l'ID dell'account a questo percorso. Ad esempio: `s3://deploy-guided-Region-AccountID/src/`.
- `cfn-name`: il nome dello CloudFormation stack. Se hai usato il valore predefinito per il nome dello stack, il tuo `cfn-name` è `sam-app`.
- `work-team-name`: Il nome della forza lavoro che hai creato quando hai creato la forza lavoro privata in SageMaker AI Ground Truth.
- `job-name-prefix`: Il prefisso per il lavoro di etichettatura SageMaker AI Ground Truth. Nota che c'è un limite di 29 caratteri per questo campo. A questo valore viene aggiunto un timestamp. Ad esempio: `my-job-name-20210902T232116`.
- `entity-types`: le entità che desideri utilizzare durante il processo di etichettatura, separate da virgole. Questo elenco deve includere tutte le entità che desideri annotare nel set di dati di addestramento.

Il lavoro di etichettatura Ground Truth visualizza solo queste entità per consentire agli annotatori di etichettare il contenuto nei documenti PDF.

Per visualizzare gli argomenti aggiuntivi supportati dallo script, utilizzate l'-hopzione per visualizzare il contenuto della guida.

- Eseguite lo script seguente con i parametri di input come descritto nell'elenco precedente.

```
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"
```

Lo script produce il seguente output:

```
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 %
```

Annotazione con SageMaker AI Ground Truth

Ora che hai configurato le risorse richieste e creato un processo di etichettatura, puoi accedere al portale di etichettatura e annotare le tue. PDFs

1. Accedi alla [console SageMaker AI](#) utilizzando i browser Web Chrome o Firefox.
2. Seleziona Etichettatura della forza lavoro e scegli Privato.
3. In Riepilogo della forza lavoro privata, seleziona l'URL di accesso al portale di etichettatura che hai creato con la tua forza lavoro privata. Accedi con le credenziali appropriate.

Se non vedi nessun lavoro elencato, non preoccuparti: l'aggiornamento può richiedere del tempo, a seconda del numero di file che hai caricato per l'annotazione.

4. Seleziona l'attività e, nell'angolo in alto a destra, scegli Inizia a lavorare per aprire la schermata di annotazione.

Vedrai uno dei tuoi documenti aperto nella schermata di annotazione e, sopra di esso, i tipi di entità che hai fornito durante la configurazione. A destra dei tipi di entità, c'è una freccia che puoi usare per navigare tra i tuoi documenti.

Instructions Shortcuts

Labeling Task: NER OFFERING_PRICE OFFERED_SHARES << 2 >>

[Table of Contents](#)

DILUTION

If you purchase units in this offering, you will experience dilution to the extent of the difference between the public offering price of the units (attributing no value to the warrants) and the net tangible book value per share of our common stock immediately after this offering.

Our net tangible book value as of June 30, 2017 was approximately \$15.9 million, or \$0.5589 per share of common stock. Net tangible book value per share is equal to our total tangible assets minus total liabilities, all divided by the number of shares of common stock outstanding as of June 30, 2017.

After giving effect to the sale of **OFFERED_SHARES** 3,265,309 units at a price of **OFFERING_PRICE** \$2.45 per unit, and after deducting our estimated placement agent fees and offering expenses payable by us, and attributing no value to the warrants, our as adjusted net tangible book value would have been approximately \$23.3 million, or approximately \$0.7357 per share of common stock, as of June 30, 2017. This represents an immediate increase in net tangible book value of approximately \$0.1768 per share to existing stockholders and an immediate dilution of approximately \$1.714 per share to new investors. The following table illustrates this calculation on a per share basis:

		OFFERING_PRICE
Public offering price per unit		\$ 2.45
Net tangible book value per share as of June 30, 2017	\$ 0.5589	
Increase per share attributable to this offering	\$ 0.1768	
As adjusted net tangible book value per share as of June 30, 2017, after giving effect to this offering		\$ 0.7357
Dilution per share to new investors		\$ 1.714

The foregoing table and discussion is based on 28,452,305 shares outstanding as of June 30, 2017 and excludes:

- 1,937,871 shares of our common stock subject to outstanding options having a weighted average exercise price of \$5.54 per share;
- 54,300 shares of our common stock subject to outstanding restricted stock units;

Annota il documento aperto. Puoi anche rimuovere, annullare o etichettare automaticamente le tue annotazioni su ogni documento; queste opzioni sono disponibili nel pannello a destra dello strumento di annotazione.

The screenshot shows a web interface for document annotation. It includes several sections: a 'METADATA' section with a text input field; a 'BLOCK' section with a 'SHOW DOCUMENT' button; a 'SELECTED ENTITIES' section with 'COPY' and 'REMOVE' buttons; an 'UNASSIGNED' section with a dropdown menu, a 'Repeat Location.' label, a range selector from 0 to 16, and a 'RESET' button; and an 'ENTITY LIST 1' section with a 'REMOVE ALL' button.

Per usare il tag auto, annota un'istanza di una delle tue entità; tutte le altre istanze di quella parola specifica vengono quindi annotate automaticamente con quel tipo di entità.

Una volta terminato, seleziona Invia in basso a destra, quindi usa le frecce di navigazione per passare al documento successivo. Ripeti l'operazione finché non avrai annotato tutti i tuoi. PDFs

Dopo aver annotato tutti i documenti di formazione, puoi trovare le annotazioni in formato JSON nel bucket Amazon S3 in questa posizione:

```
/output/your labeling job name/annotations/
```

La cartella di output contiene anche un file manifest di output, che elenca tutte le annotazioni all'interno dei documenti di formazione. È possibile trovare il file manifesto di output nella seguente posizione.

```
/output/your labeling job name/manifests/
```

Addestramento di modelli di riconoscimento di entità personalizzati

Un identificatore di entità personalizzato identifica solo i tipi di entità inclusi durante l'addestramento del modello. Non include automaticamente i tipi di entità preimpostati. Se desideri identificare anche

i tipi di entità preimpostati, come LOCATION, DATE o PERSON, devi fornire dati di formazione aggiuntivi per tali entità.

Quando si crea un riconoscimento di entità personalizzato utilizzando file PDF annotati, è possibile utilizzare il riconoscimento con una varietà di formati di file di input: testo semplice, file di immagine (JPG, PNG, TIFF), file PDF e documenti Word, senza la necessità di preelaborazione o di appiattimento del documento. Amazon Comprehend non supporta l'annotazione di file di immagine o documenti Word.

Note

Un riconoscitore di entità personalizzato che utilizza file PDF annotati supporta solo documenti in inglese.

Dopo aver creato un riconoscimento di entità personalizzato, è possibile monitorare l'avanzamento della richiesta utilizzando l'operazione. [DescribeEntityRecognizer](#) Una volta creato il Status campo TRAINED, il modello di riconoscimento è pronto all'uso per il riconoscimento personalizzato delle entità.

Argomenti

- [Addestra riconoscitori personalizzati \(console\)](#)
- [Addestra riconoscitori di entità personalizzati \(API\)](#)
- [Metriche personalizzate per il riconoscimento delle entità](#)

Addestra riconoscitori personalizzati (console)

Puoi creare riconoscitori di entità personalizzati utilizzando la console Amazon Comprehend. Questa sezione mostra come creare e addestrare un riconoscitore di entità personalizzato.

Creazione di un riconoscimento di entità personalizzato utilizzando la console - formato CSV

Per creare il riconoscimento di entità personalizzato, fornisci innanzitutto un set di dati per addestrare il tuo modello. Con questo set di dati, includi uno dei seguenti: un set di documenti annotati o un elenco di entità e la relativa etichetta di tipo, insieme a un set di documenti contenenti tali entità. Per ulteriori informazioni, consulta [Riconoscimento personalizzato delle entità](#)

Per addestrare un riconoscitore di entità personalizzato con un file CSV

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Personalizzazione, quindi scegli Riconoscimento personalizzato delle entità.
3. Scegli Crea nuovo modello.
4. Assegna un nome al riconoscitore. Il nome deve essere univoco all'interno della regione e dell'account.
5. Seleziona la lingua.
6. In Tipo di entità personalizzato, inserisci un'etichetta personalizzata che desideri che il riconoscitore trovi nel set di dati.

Il tipo di entità deve essere in lettere maiuscole e, se è composto da più di una parola, separale con un carattere di sottolineatura.

7. Scegli Aggiungi tipo.
8. Se desideri aggiungere un altro tipo di entità, inseriscilo e quindi scegli Aggiungi tipo. Se desideri rimuovere uno dei tipi di entità che hai aggiunto, scegli Rimuovi tipo, quindi scegli il tipo di entità da rimuovere dall'elenco. È possibile elencare un massimo di 25 tipi di entità.
9. Per crittografare il tuo lavoro di formazione, scegli la crittografia Recognizer, quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, per KMS key ID scegli l'ID della chiave.
 - Se utilizzi una chiave associata a un account diverso, per l'ARN della chiave KMS inserisci l'ARN per l'ID della chiave.

Note

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta. [AWS Key Management Service](#)

10. In Specifiche dei dati, scegli il formato dei tuoi documenti di formazione:
 - File CSV: un file CSV che integra i documenti di formazione. Il file CSV contiene informazioni sulle entità personalizzate che il modello addestrato rileverà. Il formato richiesto del file dipende dal fatto che stiate fornendo annotazioni o un elenco di entità.

- **Augmented manifest:** set di dati etichettato prodotto da Amazon Ground Truth SageMaker . Questo file è in formato JSON lines. Ogni riga è un oggetto JSON completo che contiene un documento di formazione e le relative etichette. Ogni etichetta annota un'entità denominata nel documento di formazione. È possibile fornire fino a 5 file manifest aumentati.

Per ulteriori informazioni sui formati disponibili e per esempi, vedere [Addestramento di modelli di riconoscimento di entità personalizzati](#).

11. In Tipo di allenamento, scegli il tipo di allenamento da utilizzare:

- Utilizzo di annotazioni e documenti di formazione
- Utilizzo dell'elenco delle entità e dei documenti di formazione

Se scegli le annotazioni, inserisci l'URL del file delle annotazioni in Amazon S3. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trovano i file di annotazione e scegliere Browse S3.

Se scegli un elenco di entità, inserisci l'URL dell'elenco di entità in Amazon S3. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trova l'elenco delle entità e scegliere Browse S3.

12. Inserisci l'URL di un set di dati di input contenente i documenti di formazione in Amazon S3. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trovano i documenti di formazione e scegliere Seleziona cartella.

13. In Set di dati di test seleziona come desideri valutare le prestazioni del tuo modello addestrato: puoi farlo sia per le annotazioni che per i tipi di addestramento con elenco di entità.

- **Autosplit:** Autosplit seleziona automaticamente il 10% dei dati di formazione forniti da utilizzare come dati di test
- **(Facoltativo) Forniti dal cliente:** quando si seleziona Fornito dal cliente, è possibile specificare esattamente quali dati di test si desidera utilizzare.

14. Se selezioni Set di dati di test fornito dal cliente, inserisci l'URL del file di annotazioni in Amazon S3. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trovano i file di annotazione e scegliere Seleziona cartella.

15. Nella sezione Scegli un ruolo IAM, seleziona un ruolo IAM esistente o creane uno nuovo.

- **Scegli un ruolo IAM esistente:** seleziona questa opzione se disponi già di un ruolo IAM con autorizzazioni per accedere ai bucket Amazon S3 di input e output.

- Crea un nuovo ruolo IAM: seleziona questa opzione quando desideri creare un nuovo ruolo IAM con le autorizzazioni appropriate per Amazon Comprehend per accedere ai bucket di input e output.

 Note

Se i documenti di input sono crittografati, il ruolo IAM utilizzato deve disporre dell'autorizzazione `kms:Decrypt`. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per utilizzare la crittografia KMS](#).

16. (Facoltativo) Per avviare le tue risorse in Amazon Comprehend da un VPC, inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
1. Scegli la sottorete in Subnet (s). Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.

 Note

Quando utilizzi un VPC con il tuo processo di riconoscimento delle entità personalizzato, l'oggetto `DataAccessRole` utilizzato per le operazioni di creazione e avvio deve disporre delle autorizzazioni per il VPC da cui si accede ai documenti di input e al bucket di output.

17. (Facoltativo) Per aggiungere un tag al riconoscimento di entità personalizzato, inserisci una coppia chiave-valore in Tag. Seleziona Aggiungi tag. Per rimuovere questa coppia prima di creare il riconoscitore, scegliete Rimuovi tag.
18. Scegli Addestra.

Il nuovo riconoscimento verrà quindi visualizzato nell'elenco, mostrandone lo stato. Inizialmente verrà visualizzato come `Submitted`. Verrà quindi visualizzato `Training` per un classificatore che sta elaborando documenti di formazione, `Trained` per un classificatore pronto all'uso e `In error` per un classificatore che presenta un errore. È possibile fare clic su un lavoro per ottenere ulteriori informazioni sul programma di riconoscimento, inclusi eventuali messaggi di errore.

Creazione di un riconoscimento di entità personalizzato utilizzando la console - manifesto aumentato

Per addestrare un riconoscitore di entità personalizzato con un documento di testo semplice, PDF o Word

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Personalizzazione, quindi scegli Riconoscimento personalizzato delle entità.
3. Scegli Train Recognizer.
4. Assegna un nome al riconoscimento. Il nome deve essere univoco all'interno della regione e dell'account.
5. Seleziona la lingua. Nota: se stai addestrando un documento PDF o Word, l'inglese è la lingua supportata.
6. In Tipo di entità personalizzato, inserisci un'etichetta personalizzata che desideri che il riconoscimento trovi nel set di dati.

Il tipo di entità deve essere in lettere maiuscole e, se è composto da più di una parola, separale con un carattere di sottolineatura.

7. Scegli Aggiungi tipo.
8. Se desideri aggiungere un altro tipo di entità, inseriscilo e quindi scegli Aggiungi tipo. Se desideri rimuovere uno dei tipi di entità che hai aggiunto, scegli Rimuovi tipo, quindi scegli il tipo di entità da rimuovere dall'elenco. È possibile elencare un massimo di 25 tipi di entità.
9. Per crittografare il tuo lavoro di formazione, scegli la crittografia Recognizer, quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, per KMS key ID scegli l'ID della chiave.
 - Se utilizzi una chiave associata a un account diverso, per l'ARN della chiave KMS inserisci l'ARN per l'ID della chiave.

Note

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta. [AWS Key Management Service](#)

10. In Dati di addestramento, scegli Augmented manifest come formato dei dati:

- **Augmented manifest:** è un set di dati etichettato prodotto da Amazon Ground Truth SageMaker . Questo file è in formato JSON lines. Ogni riga del file è un oggetto JSON completo che contiene un documento di formazione e le relative etichette. Ogni etichetta annota un'entità denominata nel documento di formazione. È possibile fornire fino a 5 file manifest aumentati. Se si utilizzano documenti PDF per i dati di addestramento, è necessario selezionare Augmented manifest. È possibile fornire fino a 5 file manifest aumentati. Per ogni file, puoi nominare fino a 5 attributi da utilizzare come dati di addestramento.

Per ulteriori informazioni sui formati disponibili e per esempi, vedere [Addestramento di modelli di riconoscimento di entità personalizzati](#).

11. Seleziona il tipo di modello di allenamento.

Se hai selezionato Documenti in testo semplice, in Posizione di input, inserisci l'Amazon S3URL del file manifesto aumentato di Amazon SageMaker AI Ground Truth. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trovano i manifesti aumentati e scegliere Seleziona cartella.

12. In Nome dell'attributo, inserisci il nome dell'attributo che contiene le tue annotazioni. Se il file contiene annotazioni provenienti da più lavori di etichettatura concatenati, aggiungi un attributo per ogni lavoro. In questo caso, ogni attributo contiene l'insieme di annotazioni di un processo di etichettatura. Nota: è possibile fornire fino a 5 nomi di attributi per ogni file.

13. Selezionare Aggiungi.

14. Se hai selezionato PDF, documenti Word in Posizione di input, inserisci l'Amazon S3URL del file manifesto aumentato di Amazon SageMaker AI Ground Truth. Puoi anche accedere al bucket o alla cartella in Amazon S3 in cui si trovano i manifesti aumentati e scegliere Seleziona cartella.

15. Inserisci il prefisso S3 per i tuoi file di dati di Annotation. Questi sono i documenti PDF che hai etichettato.

16. Inserisci il prefisso S3 per i tuoi documenti Source. Questi sono i documenti PDF originali (oggetti dati) che hai fornito a Ground Truth per il tuo lavoro di etichettatura.

17. Inserisci i nomi degli attributi che contengono le tue annotazioni. Nota: puoi fornire fino a 5 nomi di attributi per ogni file. Tutti gli attributi del file che non specifichi vengono ignorati.

18. Nella sezione ruolo IAM, seleziona un ruolo IAM esistente o creane uno nuovo.

- **Scegli un ruolo IAM esistente:** seleziona questa opzione se disponi già di un ruolo IAM con autorizzazioni per accedere ai bucket Amazon S3 di input e output.

- Crea un nuovo ruolo IAM: seleziona questa opzione quando desideri creare un nuovo ruolo IAM con le autorizzazioni appropriate per Amazon Comprehend per accedere ai bucket di input e output.

 Note

Se i documenti di input sono crittografati, il ruolo IAM utilizzato deve disporre dell'autorizzazione `kms:Decrypt`. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per utilizzare la crittografia KMS](#).

19. (Facoltativo) Per avviare le tue risorse in Amazon Comprehend da un VPC, inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
 1. Scegli la sottorete in Subnet (s). Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.

 Note

Quando utilizzi un VPC con il tuo processo di riconoscimento delle entità personalizzato, l'oggetto `DataAccessRole` utilizzato per le operazioni di creazione e avvio deve disporre delle autorizzazioni per il VPC da cui si accede ai documenti di input e al bucket di output.

20. (Facoltativo) Per aggiungere un tag al riconoscimento di entità personalizzato, inserisci una coppia chiave-valore in Tag. Seleziona Aggiungi tag. Per rimuovere questa coppia prima di creare il riconoscitore, scegliete Rimuovi tag.
21. Scegli Addestra.

Il nuovo riconoscimento verrà quindi visualizzato nell'elenco, mostrandone lo stato. Inizialmente verrà visualizzato come `Submitted`. Verrà quindi visualizzato `Training` per un classificatore che sta elaborando documenti di formazione, `Trained` per un classificatore pronto all'uso e `In error` per un classificatore che presenta un errore. È possibile fare clic su un lavoro per ottenere ulteriori informazioni sul programma di riconoscimento, inclusi eventuali messaggi di errore.

Addestra riconoscitori di entità personalizzati (API)

Per creare e addestrare un modello di riconoscimento delle entità personalizzato, utilizza l'operazione dell'API Amazon Comprehend [CreateEntityRecognizer](#)

Argomenti

- [Addestramento di riconoscitori di entità personalizzati utilizzando il AWS Command Line Interface](#)
- [Addestramento di riconoscitori di entità personalizzati utilizzando il AWS SDK per Java](#)
- [Addestramento di riconoscitori di entità personalizzati utilizzando Python \(Boto3\)](#)

Addestramento di riconoscitori di entità personalizzati utilizzando il AWS Command Line Interface

Negli esempi seguenti viene illustrato l'utilizzo dell'CreateEntityRecognizer operazione e di altre operazioni APIs associate a AWS CLI.

Gli esempi sono formattati per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Crea un riconoscimento di entità personalizzato utilizzando il comando create-entity-recognizer CLI. Per informazioni sul input-data-config parametro, consulta [CreateEntityRecognizer](#) Amazon Comprehend API Reference.

```
aws comprehend create-entity-recognizer \  
  --language-code en \  
  --recognizer-name test-6 \  
  --data-access-role-arn "arn:aws:iam::account number:role/service-role/  
AmazonComprehendServiceRole-role" \  
  --input-data-config "EntityTypes=[{Type=PERSON}],Documents={S3Uri=s3://Bucket  
Name/Bucket Path/documents},  
                        Annotations={S3Uri=s3://Bucket Name/Bucket Path/annotations}" \  
  --region region
```

Elenca tutti i riconoscitori di entità in una regione utilizzando il comando list-entity-recognizers CLI.

```
aws comprehend list-entity-recognizers \  
  --region region
```

Controlla lo stato del lavoro dei riconoscitori di entità personalizzati utilizzando il comando `describe-entity-recognizer` CLI.

```
aws comprehend describe-entity-recognizer \  
    --entity-recognizer-arn arn:aws:comprehend:region:account number:entity-  
recognizer/test-6 \  
    --region region
```

Addestramento di riconoscitori di entità personalizzati utilizzando il AWS SDK per Java

Questo esempio crea un riconoscitore di entità personalizzato e addestra il modello utilizzando Java

Per esempi di Amazon Comprehend che utilizzano Java, consulta Esempi di [Amazon Comprehend](#) Java.

Addestramento di riconoscitori di entità personalizzati utilizzando Python (Boto3)

Istanza Boto3 SDK:

```
import boto3  
import uuid  
comprehend = boto3.client("comprehend", region_name="region")
```

Crea un riconoscitore di entità:

```
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"]
```

Elenca tutti i riconoscitori:

```
response = comprehend.list_entity_recognizers()
```

Attendi che il riconoscimento raggiunga lo stato 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)
```

Metriche personalizzate per il riconoscimento delle entità

Amazon Comprehend ti fornisce metriche per aiutarti a stimare l'efficacia di un Entity Recognizer per il tuo lavoro. Si basano sull'addestramento del modello di riconoscimento e quindi, sebbene rappresentino accuratamente le prestazioni del modello durante l'addestramento, sono solo un'approssimazione delle prestazioni dell'API durante l'individuazione delle entità.

Le metriche vengono restituite ogni volta che vengono restituiti i metadati di un identificatore di entità addestrato.

Amazon Comprehend supporta l'addestramento di un modello su un massimo di 25 entità alla volta. Quando le metriche vengono restituite da un riconoscitore di entità esperto, i punteggi vengono calcolati sia per il sistema di riconoscimento nel suo insieme (metriche globali) sia per ogni singola entità (metriche di entità).

Sono disponibili tre metriche, sia come metriche globali che come metriche di entità:

- Precisione

Indica la frazione di entità prodotte dal sistema che sono correttamente identificate ed etichettate correttamente. Ciò dimostra quante volte l'identificazione dell'entità del modello è davvero una buona identificazione. È una percentuale del numero totale di identificazioni.

In altre parole, la precisione si basa su veri positivi (tp) e falsi positivi (fp) e viene calcolata come $\text{precision} = \text{tp}/(\text{tp} + \text{fp})$.

Ad esempio, se un modello prevede la presenza di due esempi di entità in un documento, dove in realtà ce n'è solo uno, il risultato è uno vero positivo e uno falso positivo. In questo caso, $\text{precision} = 1/(1 + 1)$. La precisione è del 50%, poiché una delle due entità identificate dal modello è corretta.

- **Richiama**

Indica la frazione di entità presenti nei documenti che sono correttamente identificate ed etichettate dal sistema. Matematicamente, questo valore è definito in termini di numero totale di identificazioni corrette, veri positivi (tp) e identificazioni mancate (falsi negativi).

Viene calcolato come $\text{recall} = \text{tp}/(\text{tp} + \text{fn})$. Ad esempio, se un modello identifica correttamente un'entità, ma omette altre due istanze in cui tale entità è presente, il risultato è un vero positivo e due falsi negativi. In questo caso, $\text{recall} = 1/(1 + 2)$. Il richiamo è del 33,33%, poiché un'entità è corretta tra i tre esempi possibili.

- **Punteggio F1**

Si tratta di una combinazione delle metriche Precision e Recall, che misura l'accuratezza complessiva del modello per il riconoscimento personalizzato delle entità. Il punteggio F1 è la media armonica delle metriche Precision e Recall: $F1 = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$.

 Note

Intuitivamente, la media armonica penalizza gli estremi più della media semplice o altre medie (esempio: $\text{precision} = 0$, $\text{recall} = 1$) potrebbero essere ottenute in modo banale prevedendo tutti gli intervalli possibili. Qui, la media semplice sarebbe 0,5, ma la penalizzerebbe come 0). F1

Negli esempi precedenti, $\text{precision} = 50\%$ e $\text{recall} = 33,33\%$, quindi $F1 = 2 * 0,5 * 0,3333/(0,5 + 0,3333)$. Il punteggio F1 è .3975 o 39,75%.

Metriche globali e individuali delle entità

La relazione tra le metriche globali e quelle delle singole entità può essere vista analizzando la frase seguente per entità che sono un luogo o una persona

```
John Washington and his friend Smith live in San Francisco, work in San Diego, and own a house in Seattle.
```

Nel nostro esempio, il modello effettua le seguenti previsioni.

```
John Washington = Person
Smith = Place
San Francisco = Place
San Diego = Place
Seattle = Person
```

Tuttavia, le previsioni avrebbero dovuto essere le seguenti.

```
John Washington = Person
Smith = Person
San Francisco = Place
San Diego = Place
Seattle = Place
```

Le metriche delle singole entità a tal fine sarebbero:

```
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%

Le metriche globali a tal fine sarebbero:

Globale:

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 = $2 * \text{Global Precision} * \text{Global Recall} / (\text{Global Precision} + \text{Global Recall})$)

Miglioramento delle prestazioni del riconoscimento di entità personalizzato

Queste metriche forniscono informazioni sull'accuratezza delle prestazioni del modello addestrato quando lo si utilizza per identificare le entità. Ecco alcune opzioni che puoi utilizzare per migliorare le tue metriche se sono inferiori alle tue aspettative:

1. A seconda che utilizzi [Annotazioni](#) o meno [Elenchi di entità \(solo testo semplice\)](#), assicurati di seguire le linee guida contenute nella rispettiva documentazione per migliorare la qualità dei dati. Se osservi metriche migliori dopo aver migliorato i dati e riaddestrato il modello, puoi continuare a iterare e migliorare la qualità dei dati per ottenere prestazioni migliori del modello.
2. Se utilizzi un elenco di entità, valuta la possibilità di utilizzare invece le annotazioni. Le annotazioni manuali possono spesso migliorare i risultati.

3. Se sei sicuro che non vi sia un problema di qualità dei dati e tuttavia le metriche rimangono irragionevolmente basse, invia una richiesta di assistenza.

Esecuzione di analisi di riconoscimento personalizzate in tempo reale

L'analisi in tempo reale è utile per le applicazioni che elaborano documenti di piccole dimensioni non appena arrivano. Ad esempio, puoi rilevare entità personalizzate nei post sui social media, nei ticket di assistenza o nelle recensioni dei clienti.

Prima di iniziare

È necessario un modello di riconoscimento delle entità personalizzato (noto anche come riconoscimento) prima di poter rilevare le entità personalizzate. Per ulteriori informazioni su questi modelli, vedere [the section called “Modelli di riconoscimento della formazione”](#).

Un programma di riconoscimento addestrato con annotazioni in testo semplice supporta il rilevamento delle entità solo per i documenti di testo semplice. Un programma di riconoscimento addestrato alle annotazioni dei documenti PDF supporta il rilevamento di entità per documenti di testo semplice, immagini, file PDF e documenti Word. Per informazioni sui file di input, vedere. [Ingressi per analisi personalizzate in tempo reale](#)

Se prevedi di analizzare file di immagini o documenti PDF scansionati, la tua policy IAM deve concedere le autorizzazioni per utilizzare due metodi DetectDocumentText API Amazon Textract (e). AnalyzeDocument Amazon Comprehend richiama questi metodi durante l'estrazione del testo. Per un esempio di policy, consulta [Autorizzazioni necessarie per eseguire azioni di analisi dei documenti](#).

Argomenti

- [Analisi in tempo reale per il riconoscimento personalizzato delle entità \(console\)](#)
- [Analisi in tempo reale per il riconoscimento personalizzato delle entità \(API\)](#)
- [Output per analisi in tempo reale](#)

Analisi in tempo reale per il riconoscimento personalizzato delle entità (console)

Puoi utilizzare la console Amazon Comprehend per eseguire analisi in tempo reale con un modello personalizzato. Innanzitutto, crei un endpoint per eseguire l'analisi in tempo reale. Dopo aver creato l'endpoint, esegui l'analisi in tempo reale.

Per informazioni sul provisioning della velocità effettiva degli endpoint e sui costi associati, consulta [Utilizzo degli endpoint Amazon Comprehend](#)

Argomenti

- [Creazione di un endpoint per il rilevamento personalizzato delle entità](#)
- [Esecuzione del rilevamento personalizzato delle entità in tempo reale](#)

Creazione di un endpoint per il rilevamento personalizzato delle entità

Per creare un endpoint (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Endpoints e scegli il pulsante Crea endpoint. Si apre una schermata Crea endpoint.
3. Assegna un nome all'endpoint. Il nome deve essere univoco all'interno della regione e dell'account correnti.
4. Scegli un modello personalizzato a cui desideri collegare il nuovo endpoint. Dal menu a discesa, puoi cercare per nome del modello.

Note

È necessario creare un modello prima di potervi collegare un endpoint. Se non hai ancora un modello, vedi [Addestramento di modelli di riconoscimento di entità personalizzati](#).

5. (Facoltativo) Per aggiungere un tag all'endpoint, inserisci una coppia chiave-valore in Tag e scegli Aggiungi tag. Per rimuovere questa coppia prima di creare l'endpoint, scegli Rimuovi tag.
6. Immettete il numero di unità di inferenza (IUs) da assegnare all'endpoint. Ogni unità rappresenta una velocità effettiva di 100 caratteri al secondo per un massimo di due documenti al secondo.

Per ulteriori informazioni sulla velocità effettiva degli endpoint, vedere. [Utilizzo degli endpoint Amazon Comprehend](#)

7. (Facoltativo) Se state creando un nuovo endpoint, avete la possibilità di utilizzare lo stimatore dell'interfaccia utente. Lo stimatore può aiutarti a determinare il numero di IUs richieste. Il numero di unità di inferenza dipende dalla velocità effettiva o dal numero di caratteri che si desidera analizzare al secondo.
8. Nel riepilogo degli acquisti, consulta il costo finale orario, giornaliero e mensile stimato.
9. Seleziona la casella di controllo se sai che sul tuo account vengono addebitati gli addebiti per l'endpoint dal momento in cui viene avviato fino a quando non lo elimini.
10. Seleziona Crea endpoint.

Esecuzione del rilevamento personalizzato delle entità in tempo reale

Dopo aver creato un endpoint per il modello di riconoscimento delle entità personalizzato, puoi eseguire analisi in tempo reale per rilevare le entità nei singoli documenti.

Completa i seguenti passaggi per rilevare le entità personalizzate nel testo utilizzando la console Amazon Comprehend.

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Analisi in tempo reale.
3. Nella sezione Testo di input, per Tipo di analisi, scegli Personalizzato.
4. Per Seleziona endpoint, scegli l'endpoint associato al modello di rilevamento delle entità che desideri utilizzare.
5. Per specificare i dati di input per l'analisi, puoi inserire del testo o caricare un file.
 - Per inserire del testo:
 - a. Scegli Inserisci testo.
 - b. Inserisci il testo che desideri analizzare.
 - Per caricare un file:
 - a. Scegli Carica file e inserisci il nome del file da caricare.
 - b. (Facoltativo) In Azioni di lettura avanzate, puoi sostituire le azioni predefinite per l'estrazione del testo. Per informazioni dettagliate, vedi [Impostazione delle opzioni di estrazione del testo](#).

6. Scegliete Analizza. La console mostra l'output dell'analisi, insieme a una valutazione della fiducia.

Analisi in tempo reale per il riconoscimento personalizzato delle entità (API)

Puoi utilizzare l'API Amazon Comprehend per eseguire analisi in tempo reale con un modello personalizzato. Innanzitutto, crei un endpoint per eseguire l'analisi in tempo reale. Dopo aver creato l'endpoint, esegui l'analisi in tempo reale.

Per informazioni sul provisioning della velocità effettiva degli endpoint e sui costi associati, consulta.

[Utilizzo degli endpoint Amazon Comprehend](#)

Argomenti

- [Creazione di un endpoint per il rilevamento personalizzato delle entità](#)
- [Esecuzione del rilevamento di entità personalizzate in tempo reale](#)

Creazione di un endpoint per il rilevamento personalizzato delle entità

Per informazioni sui costi associati agli endpoint, consulta. [Utilizzo degli endpoint Amazon Comprehend](#)

Creazione di un endpoint con AWS CLI

Per creare un endpoint utilizzando il AWS CLI, usa il `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 il comando ha esito positivo, Amazon Comprehend risponde con l'endpoint ARN:

```
{  
  "EndpointArn": "Arn"  
}
```

Per ulteriori informazioni su questo comando, i relativi argomenti dei parametri e il relativo output, consulta [create-endpoint](#) la sezione Command Reference. AWS CLI

Esecuzione del rilevamento di entità personalizzate in tempo reale

Dopo aver creato un endpoint per il tuo modello di riconoscimento delle entità personalizzato, utilizzi l'endpoint per eseguire l'[DetectEntities](#) operazione API. È possibile fornire un input di testo utilizzando il `text` parametro o. `bytes` Immettete gli altri tipi di input utilizzando il `bytes` parametro.

Per i file di immagine e i file PDF, è possibile utilizzare il `DocumentReaderConfig` parametro per sovrascrivere le azioni di estrazione del testo predefinite. Per informazioni dettagliate, vedi [Impostazione delle opzioni di estrazione del testo](#).

Rilevamento di entità nel testo utilizzando il AWS CLI

Per rilevare entità personalizzate nel testo, eseguite il `detect-entities` comando con il testo di input nel `text` parametro.

Example: utilizza la CLI per rilevare le entità nel testo di input

```
$ aws comprehend detect-entities \  
> --endpoint-arn arn \  
> --language-code en \  
> --text "Andy Jassy is the CEO of Amazon."
```

Se il comando ha esito positivo, Amazon Comprehend risponde con l'analisi. Per ogni entità rilevata da Amazon Comprehend, fornisce il tipo di entità, il testo, la posizione e il punteggio di affidabilità.

Rilevamento di entità in documenti semistrutturati utilizzando il AWS CLI

Per rilevare entità personalizzate in PDF, Word o file di immagine, esegui il `detect-entities` comando con il file di input nel `bytes` parametro.

Example: utilizza la CLI per rilevare le entità in un file di immagine

Questo esempio mostra come passare il file di immagine utilizzando l'`file` opzione per codificare in base64 i byte dell'immagine. Per ulteriori informazioni, consultate [Binary large objects](#) nella Guida per l'utente. AWS Command Line Interface

Questo esempio passa anche un file JSON denominato `config.json` per impostare le opzioni di estrazione del testo.

```
$ aws comprehend detect-entities \  
> --endpoint-arn arn \  
> --file image.png \  
> --language-code en \  
> --text Andy Jassy is the CEO of Amazon.
```

```
> --language-code en \  
> --bytes fileb://image1.jpg \  
> --document-reader-config file://config.json
```

Il file config.json contiene il seguente contenuto.

```
{  
  "DocumentReadMode": "FORCE_DOCUMENT_READ_ACTION",  
  "DocumentReadAction": "EXTRACT_DETECT_DOCUMENT_TEXT"  
}
```

Per ulteriori informazioni sulla sintassi dei comandi, consulta [DetectEntities](#) Amazon Comprehend API Reference.

Output per analisi in tempo reale

Uscite per input di testo

Se si inserisce del testo utilizzando il Text parametro, l'output è costituito da una serie di entità rilevate dall'analisi. L'esempio seguente mostra un'analisi che ha rilevato due entità 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"  
    }  
  ]  
}
```



```
}
```

Uscite per input semistruutturati

Per un documento di input semistruutturato o un file di testo, l'output può includere i seguenti campi aggiuntivi:

- **DocumentMetadata** — Informazioni di estrazione sul documento. I metadati includono un elenco di pagine del documento, con il numero di caratteri estratti da ciascuna pagina. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **DocumentType** — Il tipo di documento per ogni pagina del documento di input. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro.
- **Blocchi**: informazioni su ogni blocco di testo nel documento di input. I blocchi sono annidati. Un blocco di pagina contiene un blocco per ogni riga di testo, che contiene un blocco per ogni parola. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro.
- **BlockReferences** — Un riferimento a ciascun blocco per questa entità. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro. Il campo non è presente per i file di testo.
- **Errori**: errori a livello di pagina rilevati dal sistema durante l'elaborazione del documento di input. Il campo è vuoto se il sistema non ha riscontrato errori.

Per le descrizioni di questi campi di output, consulta [DetectEntities](#) Amazon Comprehend API Reference. Per ulteriori informazioni sugli elementi di layout, consulta gli [oggetti di analisi di Amazon Textract](#) nella Amazon Textract Developer Guide.

L'esempio seguente mostra l'output di un documento di input PDF scansionato di una pagina.

```
{
  "Entities": [{
    "Score": 0.9984670877456665,
    "Type": "DATE-TIME",
    "Text": "September 4,",
    "BlockReferences": [{
      "BlockId": "42dcaae-c484-4b5d-9e3f-ae0be928b3e1",
      "BeginOffset": 0,
      "EndOffset": 12,
      "ChildBlocks": [{
        "ChildBlockId": "6e9cbb43-f8be-4da0-9a4b-ff9a6c350a14",
        "BeginOffset": 0,
```

```

        "EndOffset": 9
      },
      {
        "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
        "BeginOffset": 0,
        "EndOffset": 2
      },
      {
        "ChildBlockId": "599e0d53-ae9f-491b-a762-459b22c79ff5",
        "BeginOffset": 0,
        "EndOffset": 2
      }
    ]
  ]
},
"DocumentMetadata": {
  "Pages": 1,
  "ExtractedCharacters": [{
    "Page": 1,
    "Count": 609
  }]
},
"DocumentType": [{
  "Page": 1,
  "Type": "SCANNED_PDF"
}],
"Blocks": [{
  "Id": "ee82edf3-28de-4d63-8883-40e2e4938ccb",
  "BlockType": "LINE",
  "Text": "Your Band",
  "Page": 1,
  "Geometry": {
    "BoundingBox": {
      "Height": 0.024125460535287857,
      "Left": 0.11745482683181763,
      "Top": 0.06821706146001816,
      "Width": 0.12074867635965347
    },
    "Polygon": [{
      "X": 0.11745482683181763,
      "Y": 0.06821706146001816
    },
    {
      "X": 0.2382034957408905,

```

```

        "Y": 0.06821706146001816
      },
      {
        "X": 0.2382034957408905,
        "Y": 0.09234252572059631
      },
      {
        "X": 0.11745482683181763,
        "Y": 0.09234252572059631
      }
    ]
  },
  "Relationships": [{
    "Ids": [
      "b105c561-c8d9-485a-a728-7a5b1a308935",
      "60ecb119-3173-4de2-8c5d-de182a5f86a5"
    ],
    "Type": "CHILD"
  }]
}]
}

```

L'esempio seguente mostra l'output per l'analisi di un documento PDF nativo.

Example Esempio di output tratto da un'analisi personalizzata del riconoscimento delle entità di un documento 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"
}

```

Esecuzione di processi di analisi per il riconoscimento personalizzato delle entità

È possibile eseguire un processo di analisi asincrono per rilevare entità personalizzate in un set di uno o più documenti.

Prima di iniziare

È necessario un modello di riconoscimento delle entità personalizzato (noto anche come riconoscimento) prima di poter rilevare le entità personalizzate. Per ulteriori informazioni su questi modelli, vedere [the section called “Modelli di riconoscimento della formazione”](#).

Un programma di riconoscimento addestrato con annotazioni in testo semplice supporta il rilevamento delle entità solo per i documenti di testo semplice. Un programma di riconoscimento addestrato alle annotazioni dei documenti PDF supporta il rilevamento di entità per documenti di testo semplice, immagini, file PDF e documenti Word. Per i file diversi dai file di testo, Amazon Comprehend esegue l'estrazione del testo prima di eseguire l'analisi. Per informazioni sui file di input, consulta [Ingressi per analisi asincrone personalizzate](#).

Se prevedi di analizzare file di immagini o documenti PDF scansionati, la tua policy IAM deve concedere le autorizzazioni per utilizzare due metodi DetectDocumentText API Amazon Textract (e). AnalyzeDocument Amazon Comprehend richiama questi metodi durante l'estrazione del testo. Per un esempio di policy, consulta [Autorizzazioni necessarie per eseguire azioni di analisi dei documenti](#).

Per eseguire un processo di analisi asincrona, esegui complessivamente i seguenti passaggi:

1. Archivia i documenti in un bucket Amazon S3.
2. Utilizza l'API o la console per avviare il processo di analisi.
3. Monitora lo stato di avanzamento del processo di analisi.
4. Una volta completato il processo, recupera i risultati dell'analisi dal bucket S3 che hai specificato all'avvio del lavoro.

Argomenti

- [Avvio di un processo di rilevamento delle entità personalizzato \(console\)](#)
- [Avvio di un processo di rilevamento delle entità personalizzato \(API\)](#)
- [Uscite per lavori di analisi asincroni](#)

Avvio di un processo di rilevamento delle entità personalizzato (console)

È possibile utilizzare la console per avviare e monitorare un processo di analisi asincrono per il riconoscimento personalizzato delle entità.

Per avviare un processo di analisi asincrono

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
 2. Dal menu a sinistra, scegli Analysis jobs, quindi scegli Crea job.
 3. Assegna un nome al lavoro di classificazione. Il nome deve essere unico per l'account e la regione corrente.
 4. In Tipo di analisi, scegli Riconoscimento personalizzato delle entità.
 5. Dal modello Recognizer, scegli il riconoscimento di entità personalizzato da utilizzare.
 6. Da Versione, scegli la versione del riconoscitore da utilizzare.
 7. (Facoltativo) Se scegli di crittografare i dati utilizzati da Amazon Comprehend durante l'elaborazione del lavoro, scegli Job encryption. Quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, scegli l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'ID della chiave in ARN della chiave KMS.
- 
- ### Note
- Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta [Servizio di gestione delle chiavi](#) (KMS).
8. In Dati di input, inserisci la posizione del bucket Amazon S3 che contiene i documenti di input o accedi al bucket scegliendo Browse S3. Questo bucket deve trovarsi nella stessa regione dell'API che stai chiamando. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso per il processo di analisi deve disporre delle autorizzazioni di lettura per il bucket S3.
 9. (Facoltativo) Per il formato di input, puoi scegliere il formato dei documenti di input. Il formato può essere un documento per file o un documento per riga in un singolo file. Un documento per riga si applica solo ai documenti di testo.
 10. (Facoltativo) Per la modalità di lettura dei documenti, è possibile sovrascrivere le azioni di estrazione del testo predefinite. Per ulteriori informazioni, consulta [Impostazione delle opzioni di estrazione del testo](#).

11. In Dati di output, inserisci la posizione del bucket Amazon S3 in cui Amazon Comprehend deve scrivere i dati di output del lavoro o accedervi selezionando Browse S3. Questo bucket deve trovarsi nella stessa regione dell'API che stai chiamando. Il ruolo IAM che stai utilizzando per le autorizzazioni di accesso per il processo di classificazione deve avere autorizzazioni di scrittura per il bucket S3.
12. (Facoltativo) se scegli di crittografare il risultato dell'output del tuo lavoro, scegli Crittografia. Quindi scegli se utilizzare una chiave KMS associata all'account corrente o una di un altro account.
 - Se utilizzi una chiave associata all'account corrente, scegli l'alias o l'ID della chiave per l'ID della chiave KMS.
 - Se utilizzi una chiave associata a un altro account, inserisci l'ARN per l'alias o l'ID della chiave in KMS key ID.
13. (Facoltativo) Per avviare le tue risorse in Amazon Comprehend da un VPC, inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
 1. Scegli la sottorete in Subnet (s). Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.

Note

Quando si utilizza un VPC con il lavoro di analisi, il VPC `DataAccessRole` utilizzato per le operazioni di creazione e avvio deve disporre delle autorizzazioni per il VPC che accede al bucket di output.

14. Scegli Crea lavoro per creare il processo di riconoscimento delle entità.

Avvio di un processo di rilevamento delle entità personalizzato (API)

È possibile utilizzare l'API per avviare e monitorare un processo di analisi asincrono per il riconoscimento personalizzato delle entità.

Per avviare un processo di rilevamento di entità personalizzato con

l'[StartEntitiesDetectionJob](#) operazione, fornisci il `EntityRecognizerArn`, che è l'Amazon

Resource Name (ARN) del modello addestrato. Puoi trovare questo ARN nella risposta all'[CreateEntityRecognizer](#) operazione.

Argomenti

- [Rilevamento di entità personalizzate utilizzando il AWS Command Line Interface](#)
- [Rilevamento di entità personalizzate utilizzando il AWS SDK per Java](#)
- [Rilevamento di entità personalizzate utilizzando il AWS SDK per Python \(Boto3\)](#)
- [Sovrascrivere le azioni API per i file PDF](#)

Rilevamento di entità personalizzate utilizzando il AWS Command Line Interface

Usa l'esempio seguente per ambienti Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^). Per rilevare entità personalizzate in un set di documenti, utilizzate la seguente sintassi di richiesta:

```
aws comprehend start-entities-detection-job \  
  --entity-recognizer-arn "arn:aws:comprehend:region:account number:entity-  
recognizer/test-6" \  
  --job-name infer-1 \  
  --data-access-role-arn "arn:aws:iam::account number:role/service-role/  
AmazonComprehendServiceRole-role" \  
  --language-code en \  
  --input-data-config "S3Uri=s3://Bucket Name/Bucket Path" \  
  --output-data-config "S3Uri=s3://Bucket Name/Bucket Path/" \  
  --region region
```

Amazon Comprehend risponde con JobID and JobStatus e restituirà l'output del job nel bucket S3 specificato nella richiesta.

Rilevamento di entità personalizzate utilizzando il AWS SDK per Java

Per esempi di Amazon Comprehend che utilizzano Java, consulta Esempi di [Amazon Comprehend Java](#).

Rilevamento di entità personalizzate utilizzando il AWS SDK per Python (Boto3)

Questo esempio crea un riconoscimento di entità personalizzato, addestra il modello e quindi lo esegue in un processo di riconoscimento di entità utilizzando. AWS SDK per Python (Boto3)

Crea un'istanza dell'SDK per Python.

```
import boto3
import uuid
comprehend = boto3.client("comprehend", region_name="region")
```

Crea un riconoscitore di entità:

```
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"]
```

Elenca tutti i riconoscitori:

```
response = comprehend.list_entity_recognizers()
```

Attendi che il riconoscimento dell'entità raggiunga lo stato 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)
```

Avvia un processo personalizzato di rilevamento delle entità:

```
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"  
    }  
)
```

Sovrascrivere le azioni API per i file PDF

Per i file di immagine e i file PDF, puoi sovrascrivere le azioni di estrazione predefinite utilizzando il `DocumentReaderConfig` parametro in `InputDataConfig`

L'esempio seguente definisce un file JSON denominato `myInputData Config.json` per impostare i valori. `InputDataConfig` È impostato `DocumentReadConfig` per utilizzare l'`DetectDocumentTextAPI` Amazon Textract per tutti i file 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"  
    }  
}
```

Nell'`StartEntitiesDetectionJob` operazione, specifica il file `myInputData Config.json` come parametro: `InputDataConfig`

```
--input-data-config file://myInputDataConfig.json
```

Per ulteriori informazioni sui DocumentReaderConfig parametri, vedere. [Impostazione delle opzioni di estrazione del testo](#)

Uscite per lavori di analisi asincroni

Al termine di un processo di analisi, memorizza i risultati nel bucket S3 specificato nella richiesta.

Uscite per input di testo

Per i file di input di testo, l'output è costituito da un elenco di entità per ogni documento di input.

L'esempio seguente mostra l'output di due documenti da un file di input denominato 50_docs, utilizzando un formato documento per riga.

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

```
}
```

Uscite per input semistruutturati

Per i documenti di input semistruutturati, l'output può includere i seguenti campi aggiuntivi:

- **DocumentMetadata** — Informazioni di estrazione sul documento. I metadati includono un elenco di pagine del documento, con il numero di caratteri estratti da ciascuna pagina. Questo campo è presente nella risposta se la richiesta includeva il `Byte` parametro.
- **DocumentType** — Il tipo di documento per ogni pagina del documento di input. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro.
- **Blocchi**: informazioni su ogni blocco di testo nel documento di input. I blocchi possono annidarsi all'interno di un blocco. Un blocco di pagina contiene un blocco per ogni riga di testo, che contiene un blocco per ogni parola. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro.
- **BlockReferences** — Un riferimento a ciascun blocco per questa entità. Questo campo è presente nella risposta a una richiesta che includeva il `Byte` parametro. Il campo non è presente per i file di testo.
- **Errori**: errori a livello di pagina rilevati dal sistema durante l'elaborazione del documento di input. Il campo è vuoto se il sistema non ha riscontrato errori.

Per ulteriori dettagli su questi campi di output, consulta [DetectEntities](#) Amazon Comprehend API Reference

L'esempio seguente mostra l'output di un documento di input PDF nativo di una pagina.

Example Esempio di output tratto da un'analisi di riconoscimento delle entità personalizzata di un documento PDF

```
{
  "Blocks":
  [
    {
      "BlockType": "LINE",
      "Geometry":
      {
        "BoundingBox":
        {
          "Height": 0.012575757575757575,
```

```

        "Left": 0.0,
        "Top": 0.00150631313131314,
        "Width": 0.02262091503267974
    },
    "Polygon":
    [
        {
            "X": 0.0,
            "Y": 0.00150631313131314
        },
        {
            "X": 0.02262091503267974,
            "Y": 0.00150631313131314
        },
        {
            "X": 0.02262091503267974,
            "Y": 0.0140820707070706
        },
        {
            "X": 0.0,
            "Y": 0.0140820707070706
        }
    ]
},
"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"
}

```

Creazione e gestione di modelli personalizzati

Amazon Comprehend include modelli NLP (elaborazione del linguaggio naturale) integrati che puoi utilizzare per analizzare approfondimenti o modellare argomenti. Puoi anche utilizzare Amazon Comprehend per creare modelli personalizzati per il riconoscimento delle entità e la classificazione dei documenti.

Puoi utilizzare il controllo delle versioni del modello per tenere traccia della cronologia del modello. Quando si crea e si addestra una nuova versione del modello, è possibile apportare modifiche al set di dati di addestramento. Amazon Comprehend visualizza i dettagli (incluse le prestazioni del modello) per ogni versione del modello nella pagina dei dettagli del modello. Nel tempo, puoi vedere come cambiano le prestazioni del modello man mano che apporti modifiche al set di dati di allenamento.

Puoi creare versioni del modello utilizzando la console o l'API Amazon Comprehend. In alternativa, Amazon Comprehend semplifica le attività associate alla formazione e alla valutazione di nuove versioni di modelli personalizzati. [Volani](#)

Dopo aver creato un modello personalizzato, puoi condividerlo con altri utenti consentendo ad altri di Account AWS importare una copia del tuo modello.

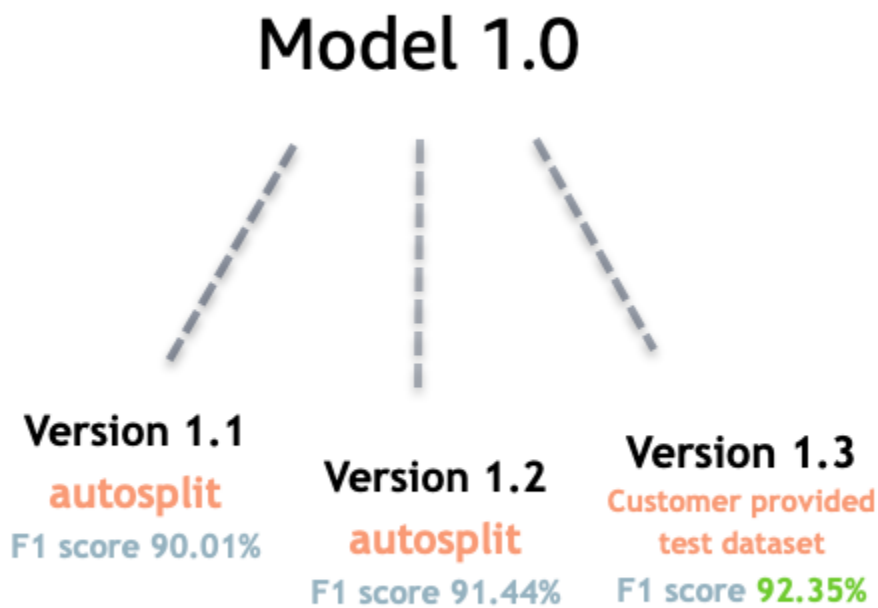
Argomenti

- [Controllo delle versioni dei modelli con Amazon Comprehend](#)
- [Copiare modelli personalizzati tra Account AWS](#)

Controllo delle versioni dei modelli con Amazon Comprehend

L'intelligenza artificiale e l'apprendimento automatico (AI/ML) si basano sulla sperimentazione rapida. Con Amazon Comprehend, puoi addestrare e creare modelli da utilizzare per ottenere informazioni dettagliate sui tuoi dati. Con il controllo delle versioni dei modelli puoi tenere traccia della cronologia di modellazione e dei punteggi associati ai risultati di esecuzione dei modelli man mano che fornisci più o più set di dati. Puoi utilizzare il controllo delle versioni con i tuoi modelli di classificazione personalizzati o i tuoi modelli di riconoscimento delle entità personalizzati. Dando un'occhiata alle diverse versioni utilizzate nel tempo, è possibile ottenere informazioni sul loro successo e sui parametri utilizzati per raggiungere lo stato di successo.

Quando addestri una nuova versione di un modello di classificatore personalizzato o di riconoscimento delle entità esistente, tutto ciò che devi fare è creare una nuova versione dalla pagina dei dettagli del modello e tutti i dettagli verranno compilati automaticamente. La nuova versione avrà lo stesso nome del modello precedente, quello che chiamiamo VersionID, anche se durante la creazione le darai un nome di versione univoco. Quando aggiungi nuove versioni a un modello, puoi vedere tutte le versioni precedenti e i relativi dettagli in un'unica visualizzazione dalla pagina dei dettagli del modello. Con il controllo delle versioni, puoi vedere come cambiano le prestazioni del modello man mano che apporti modifiche al set di dati di allenamento.



Crea una nuova versione del classificatore personalizzato (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Personalizzazione, quindi scegli Classificazione personalizzata.
3. Dall'elenco Classificatori, scegli il nome del modello personalizzato da cui desideri creare una nuova versione. Viene visualizzata la pagina dei dettagli del modello personalizzato.
4. In alto a destra, seleziona Crea nuovo modello. Si apre una schermata con i dettagli precompilati del modello di classificazione personalizzato principale.

5. In Nome versione aggiungi un nome univoco alla nuova versione.
6. In Dettagli della versione, puoi modificare la lingua e il numero di etichette associate al nuovo modello.
7. Nella sezione Specifiche dei dati, configura il modo in cui desideri fornire i dati nella nuova versione: assicurati di fornire dati completi, che includano i documenti del modello precedente e i nuovi documenti. Puoi modificare la modalità Classifier (etichetta singola o multietichetta), il formato dei dati (file CSV, manifesto aumentato), il set di dati di formazione e il set di dati di test (suddivisione automatica o configurazione personalizzata dei dati di test).
8. (Facoltativo) aggiorna la posizione S3 per i dati di output
9. In Autorizzazioni di accesso, crea o utilizza un ruolo IAM esistente.
10. (Facoltativo) Aggiorna le impostazioni del VPC
11. (Facoltativo) Aggiungi tag alla nuova versione per tenere traccia dei dettagli.

Per ulteriori informazioni sulla creazione di classificatori personalizzati, consulta [Creare un classificatore personalizzato](#)

Crea una nuova versione del riconoscimento di entità personalizzato (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Personalizzazione, quindi scegli Riconoscimento personalizzato delle entità.
3. Dall'elenco dei modelli Recognizer, scegli il nome del riconoscitore da cui desideri creare una nuova versione. Viene visualizzata la pagina dei dettagli.
4. In alto a destra, seleziona Train new version. Si apre una schermata con i dettagli precompilati del programma di riconoscimento dell'entità principale.
5. In Nome versione aggiungi un nome univoco alla nuova versione.
6. In Tipo di entità personalizzato, aggiungi le etichette o l'etichetta personalizzata che desideri che il riconoscimento identifichi nel set di dati e seleziona Aggiungi tipo. Scegli un tipo di entità personalizzato dalle annotazioni o dall'elenco di entità che hai fornito. Il sistema di riconoscimento utilizzerà quindi tutti i tipi di entità inclusi per identificare le entità nel set di dati durante l'esecuzione del processo. Ogni tipo di entità deve essere maiuscolo e separato da un carattere di sottolineatura se utilizza più parole. Sono consentiti un massimo di 25 tipi.

7. (Facoltativo) Selezionate la crittografia Recognizer per crittografare i dati nel volume di archiviazione durante l'elaborazione del lavoro.
8. Nella sezione Dati di addestramento, specifica i dettagli dell'annotazione e del formato dei dati (file CSV, manifesto aumentato) (etichetta singola o multietichetta), il formato dei dati (CSV, manifesto aumentato), il set di dati di addestramento e il set di dati di test (autosplit o configurazione personalizzata dei dati di test).
9. (Facoltativo) aggiorna la posizione S3 per i dati di output
10. In Autorizzazioni di accesso, crea o utilizza un ruolo IAM esistente.
11. (Facoltativo) Aggiorna le impostazioni del VPC
12. (Facoltativo) Aggiungi tag alla nuova versione per tenere traccia dei dettagli.

Per ulteriori informazioni sui riconoscitori di entità personalizzati, consulta [Riconoscimento di entità personalizzato e Creazione di un riconoscimento di entità personalizzato tramite la console](#).

Copiare modelli personalizzati tra Account AWS

Gli utenti di Amazon Comprehend possono copiare modelli personalizzati addestrati Account AWS in un processo in due fasi. Innanzitutto, un utente in uno Account AWS (account A) condivide un modello personalizzato presente nel proprio account. Quindi, un utente di un altro Account AWS (account B) importa il modello nel proprio account. L'utente dell'account B non ha bisogno di addestrare il modello e non ha bisogno di copiare (o accedere) ai dati di addestramento o ai dati di test originali.

Per condividere un modello personalizzato nell'account A, l'utente allega una policy AWS Identity and Access Management (IAM) a una versione del modello. Questa politica autorizza un'entità nell'account B, come un utente o un ruolo, a importare la versione del modello in Amazon Comprehend nel proprio account. Account AWS L'utente dell'account B deve importare il modello nello Regione AWS stesso modello originale.

Per importare il modello nell'account B, l'utente di questo account fornisce ad Amazon Comprehend i dettagli necessari, come l'Amazon Resource Name (ARN) del modello. Importando il modello, questo utente crea al suo interno un nuovo modello personalizzato Account AWS che replica il modello importato. Questo modello è completamente addestrato e pronto per lavori di inferenza, come la classificazione dei documenti o il riconoscimento di entità denominate.

La copia di un modello personalizzato è utile se:

- Appartieni a un'organizzazione che ne utilizza più Account AWS di uno. Ad esempio, l'organizzazione potrebbe averne una Account AWS per ogni fase di sviluppo, ad esempio compilazione, fase, test e implementazione. Oppure potrebbe avere funzioni aziendali Account AWS distinte, come la scienza dei dati e l'ingegneria.
- La tua organizzazione collabora con un altro partner, ad esempio un AWS partner, che addestra modelli personalizzati in Amazon Comprehend e te li fornisce come cliente.

In scenari come questi, puoi copiare rapidamente un identificatore di entità o un classificatore di documenti personalizzato e addestrato da uno Account AWS all'altro. Copiare un modello in questo modo è più semplice rispetto all'alternativa, in cui si copiano i dati di addestramento tra modelli Account AWS duplicati.

Argomenti

- [Condivisione di un modello personalizzato con un altro Account AWS](#)
- [Importazione di un modello personalizzato da un altro Account AWS](#)

Condivisione di un modello personalizzato con un altro Account AWS

Con Amazon Comprehend, puoi condividere i tuoi modelli personalizzati con altri, in modo che possano importare i tuoi modelli nei loro AWS account. Quando un utente importa uno dei tuoi modelli personalizzati, crea un nuovo modello personalizzato nel proprio account. Il loro nuovo modello è il duplicato di quello che hai condiviso.

Per condividere un modello personalizzato, devi allegare ad esso una politica che autorizza altri a importarlo. Quindi, fornisci a quegli utenti i dettagli di cui hanno bisogno.

Note

Quando altri utenti importano un modello personalizzato che hai condiviso, devono utilizzare lo stesso modello, ad Regione AWS esempio Stati Uniti orientali (Virginia settentrionale), che contiene il tuo modello.

Argomenti

- [Prima di iniziare](#)
- [Politiche basate sulle risorse per modelli personalizzati](#)

- [Fase 1: Aggiungere una policy basata sulle risorse a un modello personalizzato](#)
- [Passaggio 2: fornisci i dettagli che altri devono importare](#)

Prima di iniziare

Prima di poter condividere un modello, devi disporre di un classificatore personalizzato qualificato o di un riconoscimento di entità personalizzato in Amazon Comprehend nel tuo Account AWS. Per ulteriori informazioni sull'addestramento di modelli personalizzati, consulta o [Classificazione personalizzata](#) [Riconoscimento personalizzato delle entità](#)

Autorizzazioni richieste

Dichiarazione sulla politica IAM

Prima di poter aggiungere una policy basata sulle risorse a un modello personalizzato, sono necessarie le autorizzazioni in AWS Identity and Access Management (IAM). L'utente, il gruppo o il ruolo devono avere una policy allegata in modo da poter creare, ottenere ed eliminare le politiche del modello, come illustrato nell'esempio seguente.

Example Policy IAM per gestire le politiche basate sulle risorse per i modelli personalizzati

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:PutResourcePolicy",
    "comprehend>DeleteResourcePolicy",
    "comprehend:DescribeResourcePolicy"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/version/*"
}
```

Per informazioni sulla creazione di una policy IAM, consulta [Creating IAM policies nella IAM User Guide](#). Per informazioni su come allegare una policy IAM, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida](#) per l'utente IAM.

AWS KMS dichiarazione politica chiave

Se condividi un modello crittografato, potrebbe essere necessario aggiungere le autorizzazioni per AWS KMS. Questo requisito dipende dal tipo di chiave KMS che usi per crittografare il modello in Amazon Comprehend.

An Chiave di proprietà di AWS è di proprietà e gestito da un servizio. AWS Se si utilizza un Chiave di proprietà di AWS, non è necessario aggiungere le autorizzazioni per AWS KMS, e si può saltare questa sezione.

Una chiave gestita dal cliente è una chiave che puoi creare, possedere e gestire in tuo. Account AWS Se utilizzi una chiave gestita dal cliente, devi aggiungere una dichiarazione alla tua politica sulle chiavi KMS.

La dichiarazione politica autorizza una o più entità (come utenti o account) a eseguire AWS KMS le operazioni necessarie per decrittografare il modello.

Utilizzate i tasti delle condizioni per aiutare a prevenire il confuso problema del vicesceriffo. Per ulteriori informazioni, consulta [the section called “Prevenzione del confused deputy tra servizi”](#).

Utilizza le seguenti chiavi di condizione nella politica per convalidare le entità che accedono alla tua chiave KMS. Quando un utente importa il modello, AWS KMS verifica che l'ARN della versione del modello di origine corrisponda alla condizione. Se non includi una condizione nella politica, i principali specificati possono utilizzare la tua chiave KMS per decrittografare qualsiasi versione del modello:

- [aws: SourceArn](#) — Usa questa chiave di condizione con le azioni `kms:GenerateDataKey` `kms:Decrypt`
- [kms: EncryptionContext](#) — Usa questa chiave di condizione con le `kms:CreateGrant` azioni `kms:GenerateDataKey` `kms:Decrypt`, e.

Nell'esempio seguente, la politica Account AWS 444455556666 autorizza a utilizzare la versione 1 del modello di classificatore specificato di proprietà di. Account AWS 111122223333

Example Politica chiave KMS per accedere a una versione specifica del modello di classificatore

JSON

```
{
  "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"
            }
        }
    }
]
}

```

La seguente politica di esempio autorizza l'utente ExampleUser ExampleRole da Account AWS 444455556666 e verso l'accesso Account AWS 123456789012 a questa chiave KMS tramite il servizio Amazon Comprehend.

Example Politica chiave KMS per consentire l'accesso al servizio Amazon Comprehend (alternativa 1).

La seguente politica di esempio Account AWS 444455556666 autorizza l'accesso a questa chiave KMS tramite il servizio Amazon Comprehend, utilizzando una sintassi alternativa all'esempio precedente.

Example Politica chiave KMS per consentire l'accesso al servizio Amazon Comprehend (alternativa 2).

JSON

```
{
  "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:*"
        }
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Policy delle chiavi in AWS KMS](#) nella Guida per gli sviluppatori di AWS Key Management Service .

Politiche basate sulle risorse per modelli personalizzati

Prima che un utente Amazon Comprehend di un altro utente Account AWS possa importare un modello personalizzato dal tuo AWS account, devi autorizzarlo a farlo. Per autorizzarli, aggiungi una politica basata sulle risorse alla versione del modello che desideri condividere. Una policy basata sulle risorse è una policy IAM che si collega a una risorsa in AWS

Quando alleggi una policy relativa alle risorse a una versione personalizzata del modello, la policy autorizza utenti, gruppi o ruoli a eseguire `comprehend:ImportModel` sulla versione del modello.

Example Politica basata sulle risorse per una versione personalizzata del modello

Questo esempio specifica le entità autorizzate nell'attributo `Principal`. La risorsa «*» si riferisce alla versione specifica del modello a cui allegate la policy.

JSON

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

Per le politiche associate ai modelli personalizzati, `comprehend:ImportModel` è l'unica azione supportata da Amazon Comprehend.

Per ulteriori informazioni sulle politiche basate sulle risorse, consulta [Politiche basate sull'identità e politiche basate sulle risorse nella Guida per l'utente IAM](#).

Fase 1: Aggiungere una policy basata sulle risorse a un modello personalizzato

Puoi aggiungere una policy basata sulle risorse utilizzando l'API Amazon Comprehend o Amazon Console di gestione AWS Comprehend. AWS CLI

Console di gestione AWS

Puoi usare Amazon Comprehend in. Console di gestione AWS

Per aggiungere una politica basata sulle risorse

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Nel menu di navigazione a sinistra, in Personalizzazione, scegli la pagina che contiene il tuo modello personalizzato:
 - a. Se condividi un classificatore di documenti personalizzato, scegli Classificazione personalizzata.
 - b. Se condividi un riconoscimento di entità personalizzato, scegli Riconoscimento di entità personalizzato.
3. Nell'elenco dei modelli, scegli il nome del modello per aprire la relativa pagina dei dettagli.
4. In Versioni, scegli il nome della versione del modello che desideri condividere.
5. Nella pagina dei dettagli della versione, scegli la scheda Tag, VPC e policy.
6. Nella sezione Politica basata sulle risorse, scegli Modifica.
7. Nella pagina Modifica policy basata sulle risorse, procedi come segue:
 - a. Per Nome della politica, inserisci un nome che ti aiuti a riconoscere la politica dopo averla creata.
 - b. In Autorizza, specifica una o più delle seguenti entità per autorizzarle a importare il tuo modello:

Campo	Definizione ed esempi
Principali del servizio	Identificatori principali dei servizi che possono accedere a questa versione del modello. Esempio: comprehend.amazonaws.com
Account AWS IDs	Account AWS che può accedere a questa versione del modello. Autorizza tutti gli utenti che appartengono all'account. Esempio:

Campo	Definizione ed esempi
	111122223333, 123456789012
Entità IAM	ARNs per utenti o ruoli che possono accedere a questa versione del modello. Esempio: arn:aws:iam: :111122223333: user/ExampleUser, arn:aws:iam::444455556666:role/ExampleRole

8. In Condividi, puoi copiare l'ARN della versione del modello per aiutarti a condividerlo con la persona che importerà il tuo modello. Quando qualcuno importa un modello personalizzato da un altro Account AWS, è richiesta la versione del modello ARN.
9. Scegli Save (Salva). Amazon Comprehend crea la tua policy basata sulle risorse e la allega al tuo modello.

AWS CLI

Per aggiungere una politica basata sulle risorse a un modello personalizzato con, usa il comando. AWS CLI [PutResourcePolicy](#) Il comando accetta i parametri seguenti:

- **resource-arn**— L'ARN del modello personalizzato, inclusa la versione del modello.
- **resource-policy**— Un file JSON che definisce la politica basata sulle risorse da allegare al modello personalizzato.

Puoi anche fornire la policy come stringa JSON in linea. Per specificare codice JSON valido per la policy, racchiudi i nomi e i valori degli attributi tra virgolette doppie. Se anche il corpo JSON è racchiuso tra virgolette, eviti le virgolette doppie che si trovano all'interno della policy.

- **policy-revision-id**— L'ID di revisione che Amazon Comprehend ha assegnato alla politica che stai aggiornando. Se stai creando una nuova politica che non ha una versione precedente, non utilizzare questo parametro. Amazon Comprehend crea l'ID di revisione per te.

Example Aggiungi una politica basata sulle risorse a un modello personalizzato utilizzando il comando **put-resource-policy**

Questo esempio definisce una politica in un file JSON denominato Policyfile.json e la associa a un modello. Il modello è la versione v2 di un classificatore denominato 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
```

Il file JSON per la politica delle risorse contiene i seguenti contenuti:

- Azione: la politica autorizza l'uso dei principali indicati. `comprehend:ImportModel`
- Risorsa: l'ARN del modello personalizzato. La risorsa «*» si riferisce alla versione del modello specificata nel `put-resource-policy` comando.
- Principale: la politica autorizza l'utente `jane` del numero Account AWS `444455556666` e tutti gli utenti del numero `123456789012`. Account AWS

JSON

```
{  
  "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

Per aggiungere una policy basata sulle risorse a un modello personalizzato utilizzando l'API Amazon Comprehend, utilizza l'operazione API. [PutResourcePolicy](#)

Puoi anche aggiungere una policy a un modello personalizzato nella richiesta API che crea il modello. A tale scopo, fornisci la policy JSON per il ModelPolicy parametro quando invii una [CreateEntityRecognizerrichiesta](#) [CreateDocumentClassifier](#)or.

Passaggio 2: fornisci i dettagli che altri devono importare

Ora che hai aggiunto la policy basata sulle risorse al tuo modello personalizzato, hai autorizzato altri utenti di Amazon Comprehend a importare il tuo modello nel loro. Account AWS Tuttavia, prima che possano importare, devi fornire loro i seguenti dettagli:

- L'Amazon Resource Name (ARN) della versione del modello.
- Il Regione AWS che contiene il modello. Chiunque importi il tuo modello deve utilizzarlo Regione AWS .
- Se il modello è crittografato e, in caso affermativo, il tipo di AWS KMS chiave che utilizzi: Chiave di proprietà di AWS o chiave gestita dal cliente.
- Se il modello è crittografato con una chiave gestita dal cliente, è necessario fornire l'ARN della chiave KMS. Chiunque importi il tuo modello deve includere l'ARN in un ruolo di servizio IAM nel proprio. Account AWS Questo ruolo autorizza Amazon Comprehend a utilizzare la chiave KMS per decrittografare il modello durante l'importazione.

Per ulteriori informazioni su come altri utenti importano il tuo modello, consulta. [Importazione di un modello personalizzato da un altro Account AWS](#)

Importazione di un modello personalizzato da un altro Account AWS

In Amazon Comprehend, puoi importare un modello personalizzato che si trova in un altro. Account AWS Quando importi un modello, crei un nuovo modello personalizzato nel tuo account. Il tuo nuovo modello personalizzato è un duplicato completamente addestrato del modello che hai importato.

Argomenti

- [Prima di iniziare](#)
- [Importazione di un modello personalizzato](#)

Prima di iniziare

Prima di poter importare un modello personalizzato da un altro Account AWS, assicuratevi che la persona che ha condiviso il modello con voi esegua le seguenti operazioni:

- Ti autorizza a eseguire l'importazione. Questa autorizzazione è concessa nella politica basata sulle risorse allegata alla versione del modello. Per ulteriori informazioni, consulta [Politiche basate sulle risorse per modelli personalizzati](#).
- Fornisce le seguenti informazioni:
 - L'Amazon Resource Name (ARN) della versione del modello.
 - Il Regione AWS che contiene il modello. È necessario utilizzare lo stesso Regione AWS durante l'importazione.
 - Se il modello è crittografato con una AWS KMS chiave e, in caso affermativo, il tipo di chiave utilizzata.

Se il modello è crittografato, potrebbe essere necessario eseguire ulteriori passaggi, a seconda del tipo di chiave KMS utilizzata:

- Chiave di proprietà di AWS— Questo tipo di chiave KMS è di proprietà e gestito da. AWS Se il modello è crittografato con un Chiave di proprietà di AWS, non sono necessari passaggi aggiuntivi.
- Chiave gestita dal cliente: questo tipo di chiave KMS viene creata, posseduta e gestita da un AWS cliente nel proprio Account AWS. Se il modello è crittografato con una chiave gestita dal cliente, la persona che ha condiviso il modello deve:
 - Ti autorizza a decrittografare il modello. Questa autorizzazione è concessa nella politica delle chiavi KMS per la chiave gestita dal cliente. Per ulteriori informazioni, consulta [AWS KMS dichiarazione politica chiave](#).
 - Fornisci l'ARN della chiave gestita dal cliente. Utilizzi questo ARN quando crei un ruolo di servizio IAM. Questo ruolo autorizza Amazon Comprehend a utilizzare la chiave KMS per decrittografare il modello.

Autorizzazioni richieste

Prima di poter importare un modello personalizzato, tu o il tuo amministratore dovete autorizzare le azioni richieste in (IAM). AWS Identity and Access Management In qualità di utente Amazon Comprehend, devi essere autorizzato all'importazione tramite una dichiarazione di policy IAM. Se durante l'importazione è richiesta la crittografia o la decrittografia, Amazon Comprehend deve essere autorizzato a utilizzare le chiavi necessarie. AWS KMS

Dichiarazione sulla politica IAM

L'utente, il gruppo o il ruolo devono avere una policy allegata che consenta l'ImportModelazione, come mostrato nell'esempio seguente.

Example Policy IAM per importare un modello personalizzato

```
{
  "Effect": "Allow",
  "Action": [
    "comprehend:ImportModel"
  ],
  "Resource": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/foo/
version/*"
}
```

Per informazioni sulla creazione di una policy IAM, consulta [Creating IAM policies](#) nella IAM User Guide. Per informazioni su come allegare una policy IAM, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida](#) per l'utente IAM.

Ruolo di servizio IAM per la crittografia AWS KMS

Quando importi un modello personalizzato, devi autorizzare Amazon Comprehend a AWS KMS utilizzare le chiavi in uno dei seguenti casi:

- Stai importando un modello personalizzato crittografato con una chiave gestita dal cliente. AWS KMS In questo caso, Amazon Comprehend deve accedere alla chiave KMS in modo da poter decrittografare il modello durante l'importazione.
- Vuoi crittografare il nuovo modello personalizzato che crei con l'importazione e desideri utilizzare una chiave gestita dal cliente. In questo caso, Amazon Comprehend deve accedere alla tua chiave KMS per crittografare il nuovo modello.

Per autorizzare Amazon Comprehend a utilizzare AWS KMS queste chiavi, crei un ruolo di servizio IAM. Questo tipo di ruolo IAM consente a un AWS servizio di accedere alle risorse di altri servizi per tuo conto. Per ulteriori informazioni sui ruoli di servizio, consulta [Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida](#) per l'utente IAM.

Se utilizzi la console Amazon Comprehend per l'importazione, puoi fare in modo che Amazon Comprehend crei il ruolo di servizio per te. Altrimenti, devi creare un ruolo di servizio in IAM prima di importarlo.

Il ruolo di servizio IAM deve avere una politica di autorizzazioni e una politica di fiducia, come illustrato dai seguenti esempi.

Example policy di autorizzazioni

La seguente politica di autorizzazioni consente le AWS KMS operazioni utilizzate da Amazon Comprehend per crittografare e decrittografare modelli personalizzati. Garantisce l'accesso a due chiavi KMS:

- Una chiave KMS contiene il modello Account AWS da importare. È stato utilizzato per crittografare il modello e Amazon Comprehend lo utilizza per decrittografare il modello durante l'importazione.
- L'altra chiave KMS si trova nella chiave che importa il Account AWS modello. Amazon Comprehend utilizza questa chiave per crittografare il nuovo modello personalizzato creato dall'importazione.

JSON

```
{
  "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 policy di attendibilità

La seguente politica di attendibilità consente ad Amazon Comprehend di assumere il ruolo e ottenere le relative autorizzazioni. Consente al responsabile del comprehend.amazonaws.com servizio di eseguire l'`sts:AssumeRole` operazione. Per agevolare [la prevenzione della confusione](#), è possibile limitare l'ambito dell'autorizzazione utilizzando una o più chiavi di contesto relative alla condizione globale. Per `aws:SourceAccount`, specifica l'ID dell'account dell'utente che sta importando il modello.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "comprehend.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "444455556666"
        }
      }
    }
  ]
}

```

Importazione di un modello personalizzato

Puoi importare un modello personalizzato utilizzando l'API Console di gestione AWS AWS CLI, o Amazon Comprehend.

Console di gestione AWS

Puoi usare Amazon Comprehend in. Console di gestione AWS

Per importare un modello personalizzato

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Nel menu di navigazione a sinistra, in Personalizzazione, scegli la pagina per il tipo di modello che stai importando:
 - a. Se stai importando un classificatore di documenti personalizzato, scegli Classificazione personalizzata.
 - b. Se stai importando un riconoscimento di entità personalizzato, scegli Riconoscimento di entità personalizzato.
3. Scegli Importa versione.
4. Nella pagina Importa la versione del modello, inserisci i seguenti dettagli:
 - ARN della versione del modello: l'ARN della versione del modello da importare.
 - Nome modello: nome personalizzato per il nuovo modello creato dall'importazione.
 - Nome della versione: un nome personalizzato per la nuova versione del modello creata dall'importazione.
5. Per la crittografia del modello, scegli il tipo di chiave KMS da utilizzare per crittografare il nuovo modello personalizzato creato con l'importazione:
 - Usa chiave AWS proprietaria: Amazon Comprehend crittografa il tuo modello utilizzando una chiave in AWS Key Management Service (AWS KMS) creata, gestita e utilizzata per tuo conto da. AWS
 - Scegli una AWS KMS chiave diversa (avanzata): Amazon Comprehend crittografa il tuo modello utilizzando una chiave gestita dal cliente che gestisci tu stesso. AWS KMS

Se scegli questa opzione, seleziona una chiave KMS che hai o Account AWS creane una nuova scegliendo Crea una chiave. AWS KMS

6. Nella sezione Accesso al servizio, concedi ad Amazon Comprehend l'accesso a tutte AWS KMS le chiavi di cui ha bisogno per:
 - Decrittografa il modello personalizzato che importi.
 - Crittografa il nuovo modello personalizzato creato con l'importazione.

Concedi l'accesso con un ruolo di servizio IAM che consente ad Amazon Comprehend di utilizzare le chiavi KMS.

Per il ruolo di servizio, esegui una delle seguenti operazioni:

- Se disponi di un ruolo di servizio esistente che desideri utilizzare, scegli Usa un ruolo IAM esistente. Quindi, selezionalo in Nome del ruolo.
 - Se desideri che Amazon Comprehend crei un ruolo per te, scegli Crea un ruolo IAM.
7. Se hai scelto Amazon Comprehend di creare il ruolo per te, procedi come segue:
 - a. Per Nome del ruolo, inserisci un suffisso per il nome del ruolo che ti aiuterà a riconoscere il ruolo in un secondo momento.
 - b. Per ARN della chiave KMS di origine, inserisci l'ARN della chiave KMS utilizzata per crittografare il modello che stai importando. Amazon Comprehend utilizza questa chiave per decrittografare il modello durante l'importazione.
 8. (Facoltativo) Nella sezione Tag, puoi aggiungere tag al nuovo modello personalizzato che crei mediante importazione. Per ulteriori informazioni sull'etichettatura dei modelli personalizzati, consulta. [Taggare una nuova risorsa](#)
 9. Scegli Conferma.

AWS CLI

Puoi usare Amazon Comprehend eseguendo comandi con AWS CLI

Example Comando Import-model

Per importare un modello personalizzato, usa il `import-model` comando:

```
$ 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
```

Questo esempio utilizza i seguenti parametri:

- `source-model`— L'ARN del modello personalizzato da importare.
- `model-name`— Un nome personalizzato per il nuovo modello creato dall'importazione.
- `version-name`— Un nome personalizzato per la nuova versione del modello creata dall'importazione.
- `data-access-role-arn`— L'ARN del ruolo del servizio IAM che consente ad Amazon Comprehend di utilizzare le chiavi AWS KMS necessarie per crittografare o decrittografare il modello personalizzato.
- `model-kms-key-id`— L'ARN o l'ID della chiave KMS che Amazon Comprehend utilizza per crittografare il modello personalizzato creato con questa importazione. Questa chiave deve essere inserita nel tuo. AWS KMS Account AWS

API Amazon Comprehend

Per importare un modello personalizzato utilizzando l'API Amazon Comprehend, utilizza l'azione [ImportModel](#)API.

Volani

Un volano Amazon Comprehend semplifica il processo di miglioramento di un modello personalizzato nel tempo. Puoi usare un volano per orchestrare le attività associate alla formazione e alla valutazione di nuove versioni di modelli personalizzati. Flywheels supporta modelli personalizzati in testo semplice per la classificazione personalizzata e il riconoscimento personalizzato delle entità.

Argomenti

- [Panoramica del volano](#)
- [Data lake Flywheel](#)
- [Policy e autorizzazioni IAM](#)
- [Configurazione dei volani tramite la console](#)
- [Configurazione dei volani tramite l'API](#)
- [Configurazione dei set di dati](#)
- [Iterazioni del volano](#)
- [Utilizzo dei volani per l'analisi](#)

Panoramica del volano

Un flywheel è una risorsa Amazon Comprehend che orchestra la formazione e la valutazione di nuove versioni di un modello personalizzato. Puoi creare un volano per utilizzare un modello addestrato esistente oppure Amazon Comprehend può creare e addestrare un nuovo modello per il volano. Usa volani con modelli personalizzati in testo semplice per la classificazione personalizzata o il riconoscimento personalizzato delle entità.

Puoi configurare e gestire i volani utilizzando la console o l'API Amazon Comprehend. Puoi anche configurare i volani utilizzando CloudFormation.

Quando crei un flywheel, Amazon Comprehend crea un data lake nel tuo account. Il [data lake](#) archivia e gestisce tutti i dati del volano, come i dati di addestramento e i dati di test per tutte le versioni del modello.

Imposta la versione del modello attivo in modo che sia la versione del modello flywheel che desideri utilizzare per i lavori di inferenza o gli endpoint Amazon Comprehend. Inizialmente, il volano contiene una versione del modello. Nel corso del tempo, man mano che si addestrano nuove versioni del

modello, si seleziona la versione con le migliori prestazioni come versione attiva del modello. Quando un utente specifica l'ARN del flywheel per eseguire un processo di inferenza, Amazon Comprehend esegue il lavoro utilizzando la versione del modello attivo del flywheel.

Periodicamente, si ottengono nuovi dati etichettati (dati di addestramento o dati di test) per il modello. I nuovi dati vengono resi disponibili al volano creando uno o più set di dati. Un set di dati contiene dati di input per addestrare o testare il modello personalizzato associato a un volano. Amazon Comprehend carica i dati di input nel data lake del volano.

Per incorporare i nuovi set di dati nel tuo modello personalizzato, devi creare ed eseguire un'iterazione del volano. Un'iterazione con volano è un flusso di lavoro che utilizza i nuovi set di dati per valutare la versione attiva del modello e addestrare una nuova versione del modello. In base alle metriche relative alle versioni esistenti e nuove del modello, è possibile decidere se promuovere la nuova versione del modello come versione attiva.

È possibile utilizzare la versione del modello Flywheel Active per eseguire analisi personalizzate (lavori in tempo reale o asincroni). [Per utilizzare il modello del volano per l'analisi in tempo reale, è necessario creare un endpoint per il volano.](#)

Non sono previsti costi aggiuntivi per l'utilizzo dei volani. Tuttavia, quando si esegue un'iterazione del volano, si devono sostenere i costi standard per l'addestramento di una nuova versione del modello e l'archiviazione dei dati del modello. Per informazioni dettagliate sui prezzi, consulta la pagina dei prezzi di [Amazon Comprehend](#).

Argomenti

- [Set di dati Flywheel](#)
- [Creazione del volano](#)
- [Stati del volano](#)
- [Iterazioni Flywheel](#)

Set di dati Flywheel

Per aggiungere nuovi dati etichettati a un volano, si crea un set di dati. Ogni set di dati viene configurato come dati di addestramento o dati di test. Il set di dati viene associato a un volano specifico e a un modello personalizzato.

Dopo aver creato un set di dati, Amazon Comprehend carica i dati nel data lake del flywheel. Per ulteriori informazioni, consulta [Data lake Flywheel](#).

Creazione del volano

Quando create un volano, potete associare il volano a un modello addestrato esistente oppure il volano può creare un nuovo modello.

Quando create un volano con un modello esistente, specificate la versione attiva del modello. Amazon Comprehend copia i dati di addestramento e di test del modello nel data lake del volano. Assicuratevi che i dati di training e test del modello si trovino nella stessa posizione Amazon S3 in cui hai creato il modello.

Per creare un volano per un nuovo modello, fornisci un set di dati per i dati di addestramento (e un set di dati opzionale per i dati di test) quando crei il volano. Quando si esegue il volano per creare la prima iterazione del volano, il volano addestra il nuovo modello.

Quando si addestra un modello personalizzato, si specifica un elenco di etichette personalizzate (classificazione personalizzata) o entità personalizzate (riconoscimento personalizzato delle entità) che il modello deve riconoscere. Tieni presente i seguenti punti importanti sulle etichette/entità personalizzate:

- Quando create un volano per un nuovo modello, l'elenco fornito durante la creazione del labels/entities volano è l'elenco finale del volano.
- Quando create un volano da un modello esistente, l'elenco di quelli labels/entities associati a quel modello diventa l'elenco finale per il volano.
- Se si associa un nuovo set di dati al volano e tale set di dati ne contiene altri. labels/entities, Amazon Comprehend ignores the new labels/entities
- È possibile esaminare l' label/entity elenco di un volano utilizzando l'operazione API.

[DescribeFlywheel](#)

Note

Per la classificazione personalizzata, Amazon Comprehend compila l'elenco delle etichette dopo che lo stato del volano diventa ATTIVO. Attendi che il volano sia attivo prima di chiamare l'operazione API. DescribeFlywheel

Stati del volano

Un volano passa tra i seguenti stati:

- **CREAZIONE** - Amazon Comprehend sta creando le risorse Flywheel. È possibile eseguire operazioni di lettura sul volano, ad esempio. `DescribeFlywheel`
- **ATTIVO** - Il volano è attivo. È possibile determinare se è in corso un'iterazione del volano e visualizzarne lo stato. È possibile eseguire azioni di lettura sul volano e azioni come `DeleteFlywheel` `UpdateFlywheel`
- **AGGIORNAMENTO** - Amazon Comprehend sta aggiornando il volano. È possibile eseguire operazioni di lettura sul volano.
- **ELIMINAZIONE**: Amazon Comprehend sta eliminando il volano. È possibile eseguire operazioni di lettura sul volano.
- **FALLITA**: l'operazione di creazione del volano non è riuscita.

Dopo che Amazon Comprehend elimina un flywheel, conservi l'accesso a tutti i dati del modello nel data lake Flywheel. Amazon Comprehend elimina tutti i metadati interni necessari per la gestione delle risorse Flywheel. Amazon Comprehend elimina anche i set di dati associati a questo volano (i dati del modello vengono salvati nel data lake).

Iterazioni Flywheel

Quando si ottengono nuovi dati di addestramento o test per un modello di volano, si creano uno o più nuovi set di dati per caricare i nuovi dati nel data lake del volano.

Quindi esegui il volano per creare una nuova iterazione del volano. L'iterazione del volano valuta la versione corrente del modello attivo utilizzando i nuovi dati e memorizza i risultati nel data lake. Il volano crea e addestra anche una nuova versione del modello.

Se il nuovo modello presenta prestazioni migliori rispetto alla versione corrente del modello attivo, è possibile promuovere la nuova versione del modello come versione attiva del modello. È possibile utilizzare la [console](#) o il funzionamento dell'[UpdateFlywheel](#) API per aggiornare la versione attiva del modello.

Data lake Flywheel

Quando crei un volano, Amazon Comprehend crea un data lake nel tuo account per contenere tutti i dati del volano, come i dati di input e output necessari per le versioni del modello.

Amazon Comprehend crea il data lake nella posizione Amazon S3 specificata al momento della creazione del flywheel. Puoi specificare la posizione come bucket Amazon S3 o come nuova cartella in un bucket Amazon S3.

Struttura delle cartelle Data Lake

Quando Amazon Comprehend crea il data lake, configura la seguente struttura di cartelle nella posizione Amazon S3.

Warning

Amazon Comprehend gestisce l'organizzazione e i contenuti delle cartelle del data lake. Utilizza sempre le operazioni dell'API Amazon Comprehend per modificare le cartelle del data lake, altrimenti il tuo flywheel potrebbe non funzionare correttamente.

```
Document Pool
Annotations Pool
Staging
Model Datasets
  (data for each version of the model)
  VersionID-1
    Training
    Test
    ModelStats
  VersionID-2
    Training
    Test
    ModelStats
```

Per visualizzare la valutazione dell'addestramento di una versione del modello, esegui questi passaggi:

1. Aprire la cartella denominata Model Datasets al livello principale del data lake. Questa cartella contiene una sottocartella per ogni versione del modello.
2. Apri la cartella per la versione del modello che ti interessa.
3. Aprire la cartella denominata ModelStats per visualizzare le statistiche relative al modello.

Gestione del data lake

Amazon Comprehend esegue le seguenti attività per gestire il data lake per tuo conto:

- Definisce la struttura delle cartelle del data lake e inserisce i set di dati nelle cartelle appropriate.
- Gestisce i documenti di input (come file di testo e file di annotazioni) necessari per addestrare il modello.
- Gestisce i dati di output di addestramento e valutazione associati a ciascuna versione del modello.
- Gestisce la crittografia per i file archiviati nel data lake.

Amazon Comprehend esegue tutte le operazioni di creazione e aggiornamento dei dati per il data lake. Conservi l'accesso completo ai dati nel data lake. Esempio:

- Hai pieno accesso ai contenuti del data lake.
- Il data lake rimane disponibile dopo l'eliminazione del volano.
- Puoi configurare i log di accesso per il bucket Amazon S3 che contiene il data lake.
- Puoi fornire chiavi di crittografia per i dati. Queste vengono specificate quando si crea il volano.

È preferibile seguire le best practice seguenti:

- Non aggiungete manualmente le vostre cartelle o i vostri file nel data lake. Non modificare o eliminare alcun file nel data lake.
- Utilizza sempre le operazioni di creazione e aggiornamento di Amazon Comprehend per aggiungere o modificare dati nel data lake. Ad esempio, utilizzalo per `CreateDataset` fornire dati di addestramento o test e `StartFlywheelIteration` per generare dati di valutazione per le versioni del modello.
- La struttura del data lake può evolversi nel tempo. Non creare script o programmi downstream che si basino esplicitamente sulla struttura del data lake.
- Quando fornisci una posizione del data lake per il volano, ti consigliamo di creare un prefisso comune per i dati relativi a tutti i volani o di utilizzare un prefisso diverso per ogni volano. Non è consigliabile utilizzare il percorso completo del data lake di un volano come prefisso per un altro volano.

Policy e autorizzazioni IAM

Si configurano le seguenti politiche e autorizzazioni per utilizzare i volani:

- [the section called “Configura le autorizzazioni utente IAM”](#) per consentire agli utenti di accedere alle operazioni del volano.
- (Facoltativo) [the section called “Configura le autorizzazioni per le chiavi AWS KMS”](#) per il data lake.
- [the section called “Creare un ruolo di accesso ai dati”](#) che autorizza Amazon Comprehend ad accedere al data lake.

Configura le autorizzazioni utente IAM

Per utilizzare le funzionalità flywheel, aggiungi le politiche di autorizzazione appropriate alle tue identità AWS Identity and Access Management (IAM) (utenti, gruppi e ruoli).

L'esempio seguente mostra la politica di autorizzazione per creare set di dati, creare e gestire i volani e eseguire il flywheel.

Example Politica IAM per la gestione dei volani

```
{
  "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": "*"
}
```

Per informazioni sulla creazione di policy IAM per Amazon Comprehend, consulta. [Come funziona Amazon Comprehend con IAM](#)

Configura le autorizzazioni per le chiavi AWS KMS

Se utilizzi le AWS KMS chiavi per i tuoi dati nel data lake, configura le autorizzazioni richieste. Per informazioni, consulta [Autorizzazioni necessarie per utilizzare la crittografia KMS](#).

Creare un ruolo di accesso ai dati

Crei un ruolo di accesso ai dati in IAM for Amazon Comprehend per accedere ai dati Flywheel nel data lake. Se utilizzi la console per creare un volano, il sistema può facoltativamente creare un nuovo ruolo per questo scopo. Per ulteriori informazioni, consulta [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#).

Configurazione dei volani tramite la console

Puoi utilizzare la console Amazon Comprehend per creare, aggiornare ed eliminare i volani.

Quando crei un volano, Amazon Comprehend crea un data lake per contenere tutti i dati necessari al volano, come i dati di addestramento e i dati di test per ogni versione del modello.

Quando elimini un volano, Amazon Comprehend non elimina il data lake o il modello associato al volano.

Consulta le informazioni nella sezione [Creazione del volano](#) prima di creare un nuovo volano.

Argomenti

- [Create un volano](#)
- [Aggiorna un volano](#)
- [Eliminare un volano](#)

Create un volano

Quando create un volano, i campi di configurazione richiesti dipendono dal fatto che il volano sia destinato a un modello personalizzato esistente o a un nuovo modello.

Come creare un flywheel

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).

2. Dal menu a sinistra, scegli Flywheels.
3. Dalla tabella Flywheels, scegli Crea nuovo volano.
4. In Nome del volano, inserisci un nome per il volano.
5. (Facoltativo) Per creare un volano per un modello esistente, configurate i campi in Versione attiva del modello.
 - a. Dall'elenco a discesa Modello, selezionate un modello
 - b. Dall'elenco a discesa Versione, selezionare la versione del modello.
6. (Facoltativo) Per creare un nuovo modello di classificatore per il volano, in Tipo di modello personalizzato, scegliete una classificazione personalizzata e configurate i parametri nei passaggi seguenti.
 - a. In Lingua, selezionate la lingua per il modello.
 - b. In modalità Classifier, scegliete la modalità a etichetta singola o la modalità multietichetta.
 - c. In Etichette personalizzate, inserisci una o più etichette personalizzate da utilizzare per addestrare il modello. Ogni etichetta deve corrispondere a una delle classi nei dati di allenamento di input.
7. (Facoltativo) Per creare un nuovo modello di riconoscimento delle entità per il volano, in Tipo di modello personalizzato, scegli un riconoscimento delle entità personalizzato e configura i parametri nei passaggi seguenti.
 - a. In Lingua, seleziona la lingua per il modello.
 - b. In Tipo di entità personalizzato, inserisci fino a 25 entità personalizzate da utilizzare per addestrare il modello. Ogni etichetta deve corrispondere a uno dei tipi di entità nei dati di addestramento di input.

Per creare più di un'etichetta, esegui i seguenti passaggi più volte.

- i. Inserisci un'etichetta personalizzata. L'etichetta deve essere tutta in maiuscolo. Usa un carattere di sottolineatura come separatore tra le parole dell'etichetta.
- ii. Scegli Aggiungi tipo.

Per rimuovere una delle etichette che hai aggiunto, scegli X a destra del nome dell'etichetta.

8. Configura le tue scelte per la crittografia dei volumi, la crittografia dei modelli e la crittografia dei data lake. Per ognuna di queste opzioni, scegli se utilizzare una chiave KMS di tua AWS proprietà o una chiave che sei autorizzato a utilizzare.

- Se utilizzi una chiave KMS AWS di proprietà, non ci sono parametri aggiuntivi.
- Se stai usando un'altra chiave esistente, per la chiave KMS ARN inserisci l'ARN per l'ID della chiave.
- Se desideri creare una nuova chiave, scegli Crea una AWS chiave KMS.

Per ulteriori informazioni sulla creazione e l'utilizzo delle chiavi KMS e sulla crittografia associata, consulta [AWS Key Management Service](#)

- a. Configura la chiave di crittografia del volume. Amazon Comprehend utilizza questa chiave per crittografare i dati nel volume di storage durante l'elaborazione del lavoro. Scegli se utilizzare una chiave KMS di AWS proprietà o una chiave che sei autorizzato a utilizzare.
 - b. Configura la chiave di crittografia del modello. Amazon Comprehend utilizza questa chiave per crittografare i dati del modello per questa versione del modello.
9. Configura la posizione del Data lake. Per ulteriori informazioni, consulta [Gestione del data lake](#).
 10. (Facoltativo) Configura la chiave di crittografia Data lake. Amazon Comprehend utilizza questa chiave per crittografare tutti i file nel data lake.
 11. (Facoltativo) Configura le impostazioni del VPC. Inserisci l'ID VPC in VPC o scegli l'ID dall'elenco a discesa.
 1. Scegli la sottorete in Subnet (s). Dopo aver selezionato la prima sottorete, è possibile sceglierne altre.
 2. In Gruppi di sicurezza, scegli il gruppo di sicurezza da utilizzare, se ne hai specificato uno. Dopo aver selezionato il primo gruppo di sicurezza, puoi sceglierne altri.
 12. Configura le autorizzazioni di accesso al servizio.
 1. Se selezioni Usa un ruolo IAM esistente, seleziona il nome del ruolo nell'elenco a discesa.
 2. Se selezioni Crea un ruolo IAM, Amazon Comprehend crea un nuovo ruolo. La console mostra le autorizzazioni configurate da Amazon Comprehend per il ruolo. In Nome ruolo, inserisci un nome descrittivo per il ruolo.
 13. (Facoltativo) Configura le impostazioni dei tag. Per aggiungere un tag, inserisci una coppia chiave-valore in Tag. Seleziona Aggiungi tag. Per rimuovere questa coppia prima di creare il volano, scegliete Rimuovi tag. Per ulteriori informazioni, consulta [Tagging delle risorse](#).
 14. Scegli Create (Crea).

Aggiorna un volano

È possibile configurare il nome del volano, la posizione del data lake, il tipo di modello e la configurazione del modello solo quando si crea il volano.

Quando si aggiorna un volano, è possibile specificare un modello diverso se il tipo di modello e le opzioni di configurazione sono gli stessi del modello corrente. È possibile configurare una nuova versione attiva del modello. Puoi anche aggiornare i dettagli di crittografia, le autorizzazioni di accesso al servizio e le impostazioni VPC.

Per aggiornare un volano

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Flywheels.
3. Dalla tabella Flywheels, scegli il volano da aggiornare.
4. In Versione attiva del modello, scegliete un modello dall'elenco a discesa Modello e scegliete una versione del modello.

Il modulo compila il tipo di modello e la configurazione del modello.

5. (Facoltativo) Configurare le impostazioni di crittografia del volume e di crittografia del modello.
6. (Facoltativo) Configura le impostazioni di crittografia Data Lake.
7. Configura le autorizzazioni di accesso al servizio.
8. (Facoltativo) Configura le impostazioni del VPC.
9. (Facoltativo) Configura le impostazioni dei tag.
10. Scegli Save (Salva).

Eliminare un volano

Come eliminare un flywheel

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Flywheels.
3. Dalla tabella Flywheels, scegliete il volano da eliminare.
4. Scegli Delete (Elimina).

Configurazione dei volani tramite l'API

Puoi utilizzare l'API Amazon Comprehend per creare, aggiornare ed eliminare i volani.

Quando crei un volano, Amazon Comprehend crea un data lake per contenere tutti i dati necessari al volano, come i dati di addestramento e i dati di test per ogni versione del modello.

Quando elimini un volano, Amazon Comprehend non elimina il data lake o il modello associato al volano.

L'operazione di eliminazione del flywheel ha esito negativo se il flywheel sta eseguendo un'iterazione o sta creando un set di dati.

Esamina le informazioni nella sezione [Creazione del volano](#) prima di creare un nuovo volano.

Create un volano per un modello esistente

Utilizzate l'[CreateFlywheel](#) operazione per creare un volano per un modello esistente.

Example

```
aws comprehend create-flywheel \
  --flywheel-name "myFlywheel12" \
  --active-model-arn "modelArn" \
  --data-access-role-arn arn:aws::iam::111122223333:role/testFlywheelDataAccess \
  --data-lake-s3-uri": "https://s3-bucket-endpoint" \
```

Se l'operazione ha esito positivo, la risposta include l'ARN del volano.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Crea un volano per un nuovo modello

Utilizzate l'[CreateFlywheel](#) operazione per creare un volano per un nuovo modello di classificazione personalizzato.

Example

```
aws comprehend create-flywheel \
```

```
--flywheel-name "myFlywheel12" \
--data-access-role-arn arn:aws::iam::111122223333:role/testFlywheelDataAccess \
--model-type "DOCUMENT_CLASSIFIER" \
--data-lake-s3-uri "s3Uri" \
--task-config file://taskConfig.json
```

Il file TaskConfig.json contiene il seguente contenuto.

```
{
  "LanguageCode": "en",
  "DocumentClassificationConfig": {
    "Mode": "MULTI_LABEL",
    "Labels": ["optimism", "anger"]
  }
}
```

Il corpo della risposta dell'API include il seguente contenuto.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Descrivi un volano

Usa l'[DescribeFlywheel](#) operazione Amazon Comprehend per recuperare informazioni configurate su un volano.

```
aws comprehend describe-flywheel \
  --flywheel-arn "flywheelArn"
```

Il corpo della risposta dell'API include il seguente contenuto.

```
{
  "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
}
}

```

Aggiorna un volano

Utilizzate l'[UpdateFlywheel](#) operazione per aggiornare i valori di configurazione modificabili del volano.

Alcuni campi di configurazione sono strutture JSON con sottocampi. Per aggiornare uno o più sottocampi, fornisci i valori per tutti i sottocampi (Amazon Comprehend imposta il valore su null per ogni sottocampo mancante nella richiesta).

Se ometti un parametro di primo livello nella UpdateFlywheel richiesta, Amazon Comprehend non modifica i valori del parametro o dei relativi sottocampi nel volano.

Per aggiungere o rimuovere tag sul volano, usa le operazioni and. [TagResourceUntagResource](#)

È possibile promuovere una versione del modello impostando il ActiveModelArn parametro, come illustrato nell'esempio seguente.

```

aws comprehend update-flywheel \
  --region aws-region \
  --flywheel-arn "flywheelArn" \
  --active-model-arn "modelArn" \

```

Il corpo della risposta dell'API include il seguente contenuto.

```

{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}

```

Eliminare un volano

Usa l'[DeleteFlywheel](#) operazione Amazon Comprehend per eliminare i volani.

```
aws comprehend delete-flywheel \  
  --flywheel-arn "flywheelArn"
```

Una risposta API riuscita contiene un corpo del messaggio di risposta vuoto

Elenca i volani

Usa l'[ListFlywheels](#) operazione Amazon Comprehend per recuperare un elenco di volani nella regione corrente.

```
aws comprehend list-flywheel \  
  --region aws-region \  
  --endpoint-url "uri"
```

Il corpo della risposta dell'API include il seguente contenuto.

```
{  
  "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  
    }  
  ]  
}
```

Configurazione dei set di dati

Per aggiungere dati di addestramento o test etichettati a un volano, usa la console o l'API Amazon Comprehend per creare un set di dati.

Puoi configurare ogni set di dati come dati di addestramento o dati di test. Il set di dati viene associato a un volano specifico e a un modello personalizzato. Quando crei un set di dati, Amazon Comprehend carica i dati nel data lake del flywheel. Per dettagli sui formati di file per i dati di

addestramento, consulta o. [Preparazione dei dati di addestramento del classificatore](#) [Preparazione dei dati di formazione di Entity Recognizer](#)

Quando elimini il flywheel, Amazon Comprehend elimina i set di dati. I dati caricati rimangono disponibili nel data lake.

Creazione di un set di dati (console)

Creazione di un set di dati

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Flywheels e scegli il volano a cui desideri aggiungere i dati.
3. Scegli la scheda Datasets.
4. Nella tabella Set di dati di addestramento o Set di dati di test, scegli Crea set di dati.
5. In Dettagli del set di dati, inserisci un nome per il set di dati e una descrizione opzionale.
6. In Specifiche dei dati, scegli i campi Formato dati e Configurazione del tipo di set di dati.
7. (Facoltativo) In Formato di input, scegliete il formato dei documenti di input.
8. In Posizione delle annotazioni su S3, inserisci la posizione Amazon S3 del file delle annotazioni.
9. In Posizione dei dati di formazione su S3, inserisci la posizione Amazon S3 dei file dei documenti.
10. Scegli Create (Crea).

Creazione di un set di dati (API)

È possibile utilizzare l'[CreateDataset](#) operazione per creare un set di dati.

Example

```
aws comprehend create-dataset \  
  --flywheel-arn "myFlywheel12" \  
  --dataset-name "my-training-dataset" \  
  --dataset-type "TRAIN" \  
  --description "my training dataset" \  
  --cli-input-json file://inputConfig.json \  
}
```

Il file `inputConfig.json` contiene il seguente contenuto.

```
{
  "DataFormat": "COMPREHEND_CSV",
  "DocumentClassifierInputDataConfig": {
    "S3Uri": "s3://my-comprehend-datasets/multilabel_train.csv"
  }
}
```

Per aggiungere o rimuovere tag sul set di dati, usa le operazioni [TagResource](#) and [UntagResource](#).

Descrivi un set di dati

Usa l'[DescribeDataset](#) operazione Amazon Comprehend per recuperare informazioni configurate su un volano.

```
aws comprehend describe-dataset \
  --dataset-arn "datasetARN"
```

La risposta contiene il seguente contenuto.

```
{
  "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
  }
}
```

Iterazioni del volano

Usa le iterazioni Flywheel per aiutarti a creare e gestire nuove versioni del modello.

Argomenti

- [Flusso di lavoro di iterazione](#)

- [Gestione delle iterazioni \(console\)](#)
- [Gestione delle iterazioni \(API\)](#)

Flusso di lavoro di iterazione

Un volano inizia con una versione del modello addestrata o utilizza un set di dati iniziale per addestrare una versione del modello.

Nel tempo, man mano che si ottengono nuovi dati etichettati, si addestrano nuove versioni del modello per migliorare le prestazioni del modello di volano. Quando si esegue il volano, viene creata una nuova iterazione che addestra e valuta una nuova versione del modello. È possibile promuovere la nuova versione del modello se le sue prestazioni sono superiori alla versione attiva del modello esistente.

Il flusso di lavoro di iterazione del flywheel include i seguenti passaggi:

1. Crei set di dati per i nuovi dati etichettati.
2. Si esegue il volano per creare una nuova iterazione. L'iterazione segue questi passaggi per addestrare e valutare una nuova versione del modello:
 - a. Valuta la versione attiva del modello utilizzando i nuovi dati.
 - b. Addestra una nuova versione del modello utilizzando i nuovi dati.
 - c. Memorizza i risultati della valutazione e della formazione nel data lake.
 - d. Restituisce i punteggi F1 per entrambi i modelli.
3. Al termine dell'iterazione, è possibile confrontare i punteggi F1 per il modello attivo esistente e il nuovo modello.
4. Se la nuova versione del modello offre prestazioni superiori, la promuovete come versione attiva del modello. È possibile utilizzare la [console](#) o l'[API](#) per promuovere la nuova versione del modello.

Gestione delle iterazioni (console)

È possibile utilizzare la console per avviare una nuova iterazione e interrogare lo stato di un'iterazione in corso. È inoltre possibile visualizzare i risultati delle iterazioni completate.

Avvia un'iterazione con volano (console)

Prima di iniziare una nuova iterazione, create uno o più nuovi set di dati di addestramento o test. Per informazioni, consultare [Configurazione dei set di dati](#).

Avvia un'iterazione del volano (console)

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Flywheels.
3. Dalla tabella Flywheels, scegli un volano.
4. Scegliete Run flywheel.

Analizza i risultati dell'iterazione (console)

Dopo aver eseguito l'iterazione del volano, la console visualizza i risultati nella tabella delle iterazioni Flywheels.

Promuovi una nuova versione del modello (Console)

Dalla pagina dei dettagli del modello nella console, è possibile promuovere una nuova versione del modello come versione attiva del modello.

Promuovi una versione del modello Flywheel alla versione attiva del modello (console)

1. Accedi Console di gestione AWS e apri la console [Amazon Comprehend](#).
2. Dal menu a sinistra, scegli Flywheels.
3. Dalla tabella Flywheels, scegli un volano.
4. Dalla tabella della pagina dei dettagli di Flywheel, scegli la versione da promuovere dalla tabella delle iterazioni di Flywheels.
5. Scegliete Crea modello attivo.

Gestione delle iterazioni (API)

Puoi utilizzare l'API Amazon Comprehend per iniziare una nuova iterazione e interrogare lo stato di un'iterazione in corso. Puoi anche visualizzare i risultati delle iterazioni completate.

Avvia l'iterazione del volano (API)

Usa l'[StartFlywheelIteration](#) operazione Amazon Comprehend per avviare un'iterazione del volano.

```
aws comprehend start-flywheel-iteration \  
  --flywheel-arn "flywheelArn"
```

La risposta contiene il seguente contenuto.

```
{
  "FlywheelIterationArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name"
}
```

Promuovi una nuova versione del modello (API)

Utilizzate l'[UpdateFlywheel](#) operazione per promuovere una versione del modello come versione attiva del modello.

Invia la `UpdateFlywheel` richiesta con il `ActiveModelArn` parametro impostato all'ARN della nuova versione attiva del modello.

```
aws comprehend update-flywheel \
  --active-model-arn "modelArn" \
```

La risposta contiene il seguente contenuto.

```
{
  "FlywheelArn": "arn:aws::comprehend:aws-region:111122223333:flywheel/name",
  "ActiveModelArn": "modelArn"
}
```

Descrivi i risultati dell'iterazione del volano (API)

L'[DescribeFlywheelIteration](#) operazione Amazon Comprehend restituisce informazioni su un'iterazione dopo che è stata completata.

```
aws comprehend describe-flywheel-iteration \
  --flywheel-arn "flywheelArn" \
  --flywheel-iteration-id "flywheelIterationId" \
  --region aws-region
```

La risposta contiene i seguenti contenuti.

```
{
  "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>
    }
}
}
}

```

Ottieni la cronologia delle iterazioni (API)

Usa l'[ListFlywheelIterationHistory](#) operazione per ottenere informazioni sulla cronologia delle iterazioni.

```

aws comprehend list-flywheel-iteration-history \
  --flywheel-arn "flywheelArn"

```

La risposta contiene il seguente contenuto.

```

{
  "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
    }
}
]
}

```

Utilizzo dei volani per l'analisi

È possibile utilizzare la versione del modello attivo del volano per eseguire analisi per la classificazione personalizzata o il riconoscimento delle entità. La versione attiva del modello è configurabile. È possibile utilizzare la [console](#) o l'operazione [UpdateFlywheel](#) API per impostare una nuova versione del modello come versione attiva del modello.

Per utilizzare il volano, specificate l'ARN del volano invece di un modello ARN personalizzato quando configurate l'attività di analisi. Amazon Comprehend esegue l'analisi utilizzando la versione del modello attivo del volano.

Analisi in tempo reale

Si utilizza un endpoint per eseguire analisi in tempo reale. Quando si crea o si aggiorna un endpoint, è possibile configurarlo con l'ARN del flywheel anziché con un ARN modello. Quando eseguite l'analisi in tempo reale, selezionate l'endpoint associato al volano. Amazon Comprehend esegue l'analisi utilizzando la versione del modello attivo del volano.

Quando imposti una nuova versione attiva del modello per il flywheel, l'endpoint si aggiorna automaticamente per iniziare a utilizzare la nuova versione attiva del modello. [UpdateFlywheel](#) Se non desideri che l'endpoint si aggiorni automaticamente, configura l'endpoint (utilizzando [UpdateEndpoint](#)) per utilizzare direttamente la versione del modello ARN. L'endpoint continua a utilizzare questa versione del modello se la versione del modello Flywheel Active cambia.

Per una classificazione personalizzata, utilizzate l'operazione [ClassifyDocument](#) API. Per il riconoscimento personalizzato delle entità, utilizza la richiesta [DetectEntities](#) API. Fornisci il punto finale del volano nel `EndpointArn` parametro.

Puoi anche utilizzare la console per eseguire analisi in tempo reale per la [classificazione personalizzata](#) o il riconoscimento [personalizzato delle entità](#).

Lavori asincroni

Per una classificazione personalizzata, utilizza la richiesta [StartDocumentClassificationJob](#) API per avviare un processo asincrono. Fornisci il `FlywheelArn` parametro anziché `DocumentClassifierArn`.

Per il riconoscimento personalizzato delle entità, utilizza la richiesta [StartEntitiesDetectionJob](#) API. Fornisci il `FlywheelArn` parametro anziché `EntityRecognizerArn`.

[È possibile utilizzare la console per eseguire processi di analisi asincroni per la classificazione personalizzata o il riconoscimento personalizzato delle entità.](#) Quando create il lavoro, inserite l'ARN del volano nel campo `Recognizer model` o `Classifier model`.

Gestione degli endpoint Amazon Comprehend

In Amazon Comprehend, gli endpoint rendono disponibili i modelli personalizzati per la classificazione in tempo reale o il rilevamento di entità. Dopo aver creato un endpoint, puoi modificarlo man mano che le tue esigenze aziendali si evolvono. Ad esempio, è possibile monitorare l'utilizzo degli endpoint e applicare la scalabilità automatica per impostare automaticamente il provisioning degli endpoint in base alle proprie esigenze di capacità. È possibile gestire tutti gli endpoint da un'unica visualizzazione e, quando non è più necessario un endpoint, eliminarlo per risparmiare sui costi.

Prima di poter gestire un endpoint, devi crearne uno. Per ulteriori informazioni, consulta le procedure seguenti:

- [Creazione di un endpoint per la classificazione personalizzata](#)
- [Creazione di un endpoint per il rilevamento personalizzato delle entità](#)

Argomenti

- [Panoramica degli endpoint Amazon Comprehend](#)
- [Utilizzo degli endpoint Amazon Comprehend](#)
- [Monitoraggio degli endpoint Amazon Comprehend](#)
- [Aggiornamento degli endpoint Amazon Comprehend](#)
- [Utilizzo Trusted Advisor con Amazon Comprehend](#)
- [Eliminazione degli endpoint Amazon Comprehend](#)
- [Scalabilità automatica con endpoint](#)

Panoramica degli endpoint Amazon Comprehend

La pagina degli endpoint della console Amazon Comprehend ti offre una visione globale dei tuoi endpoint. Dalla pagina di panoramica degli endpoint, puoi visualizzare tutti gli endpoint in un unico posto per comprendere l'utilizzo degli endpoint rispetto all'utilizzo effettivo delle risorse. In alto a destra della pagina degli endpoint puoi specificare quali endpoint vuoi visualizzare: tutti, endpoint di classificazione personalizzati o endpoint di entità personalizzati.

Puoi creare, aggiornare, monitorare ed eliminare gli endpoint da questa pagina. Dalla sezione panoramica degli endpoint, puoi visualizzare un elenco degli endpoint, i modelli personalizzati

ospitati dagli endpoint, l'ora di creazione, il throughput assegnato e lo stato dell'endpoint. Quando si seleziona un endpoint specifico dalla tabella di panoramica degli endpoint, vengono visualizzati i dettagli dell'endpoint.

Inoltre, se sei un cliente [AWS Business Support](#) o [AWS Enterprise Support](#), hai accesso ai controlli Trusted Advisor specifici per i tuoi endpoint. Per ulteriori informazioni, consulta [Utilizzo Trusted Advisor con Amazon Comprehend](#). Per un elenco completo dei controlli e delle descrizioni, consulta le [best practice di Trusted Advisor](#).

Per ulteriori informazioni sulla gestione degli endpoint, consulta i seguenti argomenti.

- [Utilizzo degli endpoint Amazon Comprehend](#)
- [Monitoraggio degli endpoint Amazon Comprehend](#)
- [Aggiornamento degli endpoint Amazon Comprehend](#)
- [Utilizzo Trusted Advisor con Amazon Comprehend](#)
- [Eliminazione degli endpoint Amazon Comprehend](#)

Important

Il costo della classificazione personalizzata in tempo reale si basa sia sulla velocità effettiva impostata che sul periodo di tempo in cui l'endpoint è attivo. Se non si utilizza più l'endpoint o non lo si utilizza per un periodo prolungato, è necessario impostare una politica di scalabilità automatica per ridurre i costi. In alternativa, se non utilizzi più un endpoint, puoi eliminarlo per evitare costi aggiuntivi. Per ulteriori informazioni, consulta [Scalabilità automatica con endpoint](#).

Utilizzo degli endpoint Amazon Comprehend

Si crea un endpoint per eseguire analisi in tempo reale utilizzando un modello personalizzato. Un endpoint include risorse gestite che rendono disponibile il modello personalizzato per l'inferenza in tempo reale.

Amazon Comprehend assegna la velocità effettiva a un endpoint utilizzando unità di inferenza (UI). Un'interfaccia utente rappresenta un throughput di dati di 100 caratteri al secondo. È possibile fornire all'endpoint fino a 10 unità di inferenza. È possibile aumentare o diminuire il throughput dell'endpoint aggiornando l'endpoint.

Se i documenti di input includono documenti semistruutturati o file di immagine, la velocità effettiva di 100 caratteri al secondo si riferisce ai caratteri estratti dal file di input. Il numero di caratteri IUs forniti per un endpoint dipende dalla densità di caratteri dei documenti di input.

Le risposte [ClassifyDocument](#) e [DetectEntities](#) API includono il conteggio dei caratteri per ogni pagina di input. È possibile utilizzare queste informazioni per stimare il numero di unità di inferenza da fornire per ottenere il throughput desiderato.

Dopo aver completato l'analisi in tempo reale, elimina l'endpoint perché il relativo addebito continua finché è attivo. Puoi creare un altro endpoint quando sei pronto per eseguire ulteriori analisi in tempo reale.

Per ulteriori informazioni sul costo degli endpoint, consulta la pagina dei prezzi di [Amazon Comprehend](#).

Dopo aver creato un endpoint, puoi monitorarlo con Amazon CloudWatch, aggiornarlo per modificarne le unità di inferenza o eliminarlo quando non è più necessario. Per ulteriori informazioni, consulta [Monitoraggio degli endpoint Amazon Comprehend](#).

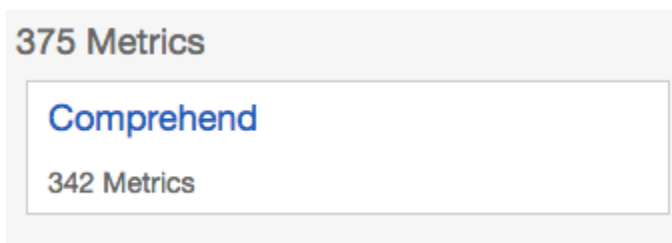
Monitoraggio degli endpoint Amazon Comprehend

È possibile regolare la velocità effettiva dell'endpoint aumentando o diminuendo il numero di unità di inferenza (). IUs Per ulteriori informazioni sull'aggiornamento dell'endpoint, consulta [the section called “Aggiornamento degli endpoint”](#)

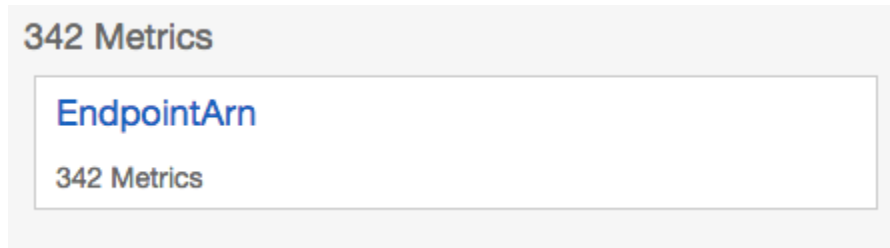
Puoi determinare come regolare al meglio il throughput del tuo endpoint monitorandone l'utilizzo con la console Amazon CloudWatch .

Monitora l'utilizzo degli endpoint con CloudWatch

1. Accedi a Console di gestione AWS e apri la [CloudWatch console](#).
2. A sinistra, scegli Metriche e seleziona Tutte le metriche.
3. In Tutte le metriche, scegli Comprehend.



4. La CloudWatch console visualizza le dimensioni per le metriche Comprehend. Scegli la dimensione. EndpointArn



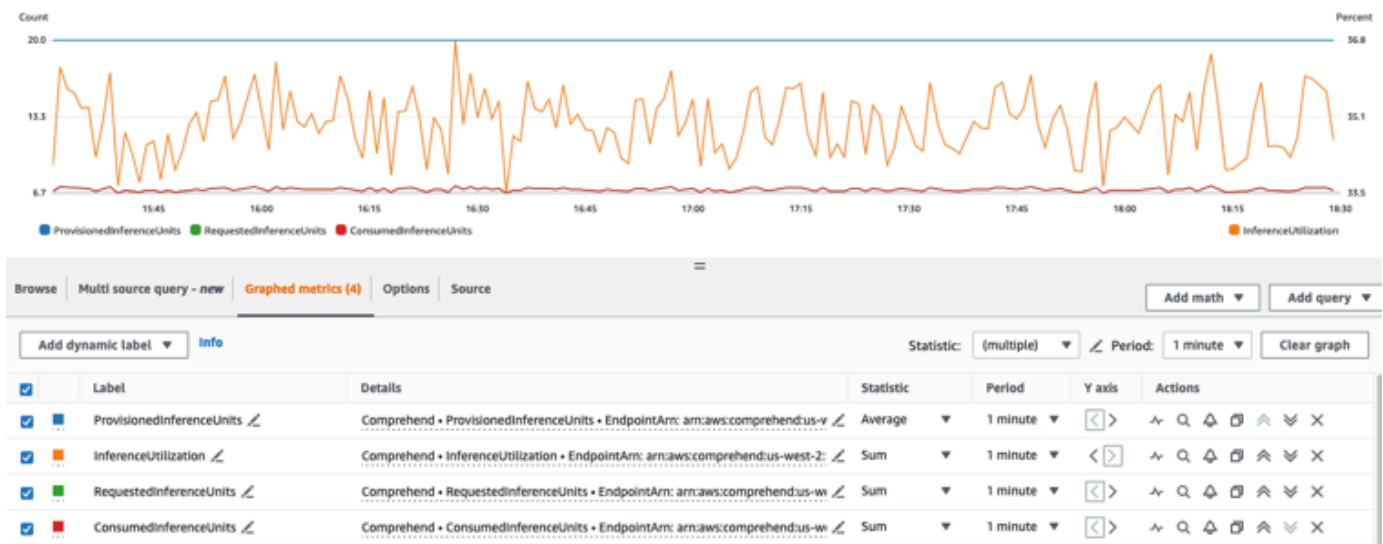
La console mostra ProvisionedInferenceUnits, RequestedInferenceUnitsConsumedInferenceUnits, e InferenceUtilizationper ciascuno dei tuoi endpoint.

Metric name ▾
<u>ProvisionedInferenceUnits</u>
<u>RequestedInferenceUnits</u>
<u>ConsumedInferenceUnits</u>
<u>InferenceUtilization</u>

Seleziona le quattro metriche e vai alla scheda Metriche grafiche.

5. Imposta le colonne delle statistiche per RequestedInferenceUnitse su Sum.
ConsumedInferenceUnits
6. Imposta la colonna Statistica InferenceUtilizationper Sum.
7. Imposta la colonna Statistica ProvisionedInferenceUnitsper su Media.
8. Modifica la colonna Periodo per tutte le metriche su 1 minuto.
9. Seleziona InferenceUtilizatione seleziona la freccia per spostarla su un asse Y separato.

Il grafico è pronto per l'analisi.



In base alle CloudWatch metriche, puoi anche impostare la scalabilità automatica per regolare automaticamente il throughput del tuo endpoint. Per ulteriori informazioni sull'utilizzo della scalabilità automatica con gli endpoint, consulta [Scalabilità automatica con endpoint](#)

- **ProvisionedInferenceUnits**- Questa metrica rappresenta il numero di risorse medie fornite IUs al momento della richiesta.
- **RequestedInferenceUnits**- Si basa sull'utilizzo di ogni richiesta inviata al servizio che è stata inviata per essere elaborata. Questo può essere utile per confrontare la richiesta inviata per essere elaborata con quella effettivamente elaborata senza ricevere throttling (**ConsumedInferenceUnits**). Il valore di questa metrica viene calcolato prendendo il numero di caratteri inviati per l'elaborazione e dividendolo per il numero di caratteri che possono essere elaborati in un minuto per 1 UI.
- **ConsumedInferenceUnits**- Si basa sull'utilizzo di ogni richiesta inviata al servizio che è stata elaborata con successo (non limitata). Questo può essere utile quando confronti ciò che stai consumando con quello che ti è stato fornito. IUs Il valore di questa metrica viene calcolato prendendo il numero di caratteri elaborati e dividendolo per il numero di caratteri che possono essere elaborati in un minuto per 1 UI.
- **InferenceUtilization**- Viene emesso per richiesta. Questo valore viene calcolato prendendo il consumo IUs definito in **ConsumedInferenceUnitse** dividendolo per **ProvisionedInferenceUnitse** convertendolo in una percentuale su 100.

 Note

Tutte le metriche vengono emesse solo per le richieste riuscite. La metrica non verrà visualizzata se proviene da una richiesta limitata o non riuscita a causa di un errore interno del server o di un errore del cliente.

Aggiornamento degli endpoint Amazon Comprehend

Spesso, il livello di throughput necessario cambia dopo la creazione di un endpoint o la prima stima delle esigenze cambia. Quando ciò accade, potrebbe essere necessario aggiornare l'endpoint per aumentare o ridurre il throughput. La velocità effettiva è regolata dal numero di unità di inferenza con cui hai fornito l'endpoint. Ogni unità di inferenza rappresenta una velocità effettiva di 100 caratteri al secondo per un massimo di 2 documenti al secondo. Potresti anche voler aggiornare la versione del modello associato all'endpoint. Quando modifichi un endpoint, puoi scegliere una versione diversa del modello per l'endpoint.

Può anche essere utile aggiungere tag all'endpoint per mantenerlo organizzato. Questa operazione può essere eseguita anche durante l'aggiornamento dell'endpoint. Per ulteriori informazioni sugli endpoint, consulta [Tagging delle risorse](#)

Per aggiornare un endpoint (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Endpoints.
3. Dall'elenco dei classificatori, scegli il nome del modello personalizzato da cui desideri aggiornare l'endpoint e segui il link. Viene visualizzata la pagina dei dettagli del modello.
4. Dalla pagina dei dettagli del modello, selezionare i dettagli della versione. Viene visualizzato l'elenco degli endpoint.
5. Seleziona la casella di controllo dell'endpoint per il tuo endpoint. In alto a destra nella tabella degli endpoint, seleziona l'icona Azioni.
6. Scegli Modifica. È possibile aggiornare i tag assegnati IUs e modificare.
7. Salvare le modifiche.
8. Per modificare il numero di unità di inferenza con cui viene fornito l'endpoint, scegli Modifica.

9. Immettete il numero aggiornato di unità di inferenza da assegnare all'endpoint. Ogni unità rappresenta una velocità effettiva di 100 caratteri al secondo. È possibile assegnare fino a un massimo di 10 unità di inferenza per endpoint.

Note

Il costo dell'utilizzo di un endpoint si basa sulla quantità di tempo operativo e sulla velocità effettiva (basata sul numero di unità di inferenza). L'aumento del numero di unità di inferenza aumenterà quindi il costo operativo. Per ulteriori informazioni, consulta i prezzi di [Amazon Comprehend](#).

10. Scegli Modifica endpoint. Viene visualizzata la pagina dei dettagli dell'endpoint.
11. Conferma che l'endpoint si stia aggiornando scegliendo il nome del modello dai breadcrumb nella parte superiore della pagina. Nella pagina dei dettagli del modello personalizzato, accedi all'elenco degli endpoint e verifica che accanto all'endpoint sia presente la dicitura Aggiornamento. Una volta completato l'aggiornamento, verrà visualizzato il messaggio Pronto.

L'esempio seguente dimostra l'utilizzo dell'UpdateEndpointoperazione con la AWS CLI.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```
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 l'azione ha esito positivo, Amazon Comprehend risponde con una risposta HTTP 200 con un corpo HTTP vuoto.

12. Per modificare il modello personalizzato collegato al tuo endpoint, dalla pagina dei dettagli del modello personalizzato, accedi all'elenco degli endpoint.
13. Seleziona l'endpoint che desideri modificare e seleziona Modifica.
14. Dalla pagina delle impostazioni dell'endpoint, in Seleziona il modello di classificatore o Seleziona il modello di riconoscimento a seconda dell'endpoint, puoi cercare un modello nel menu a discesa. Seleziona il modello che desideri.

15. In Seleziona versione puoi cercare la versione del modello che desideri. Seleziona la versione.
16. Seleziona Modifica endpoint per salvare.

Utilizzo Trusted Advisor con Amazon Comprehend

AWS Trusted Advisor è uno strumento online che fornisce consigli per aiutarvi a fornire le vostre risorse seguendo le AWS migliori pratiche.

Se disponi di un piano Basic o Developer Support, puoi utilizzare la Trusted Advisor console per accedere a tutti i controlli nella categoria Service Limits e a sei controlli nella categoria Security. Se disponi di un piano di supporto Business o Enterprise, puoi utilizzare la Trusted Advisor console e l'[Supporto AWS API](#) per accedere a tutti i Trusted Advisor controlli.

Amazon Comprehend supporta i seguenti Trusted Advisor controlli per aiutare i clienti a ottimizzare i costi e la sicurezza degli endpoint Amazon Comprehend fornendo consigli pratici.

Endpoint Amazon Comprehend sottoutilizzati

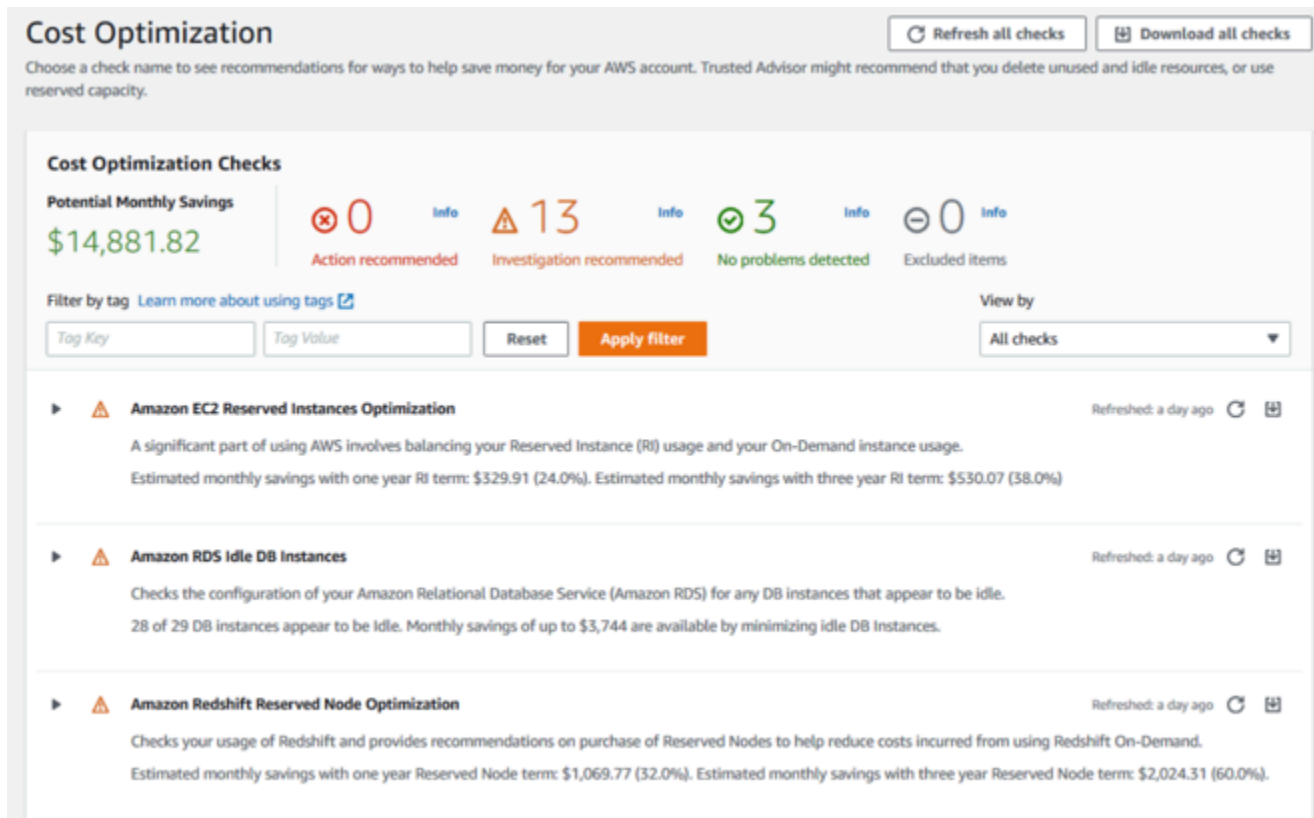
Il controllo degli endpoint sottoutilizzati di Amazon Comprehend valuta la configurazione del throughput degli endpoint. Questo controllo ti avvisa quando gli endpoint non vengono utilizzati attivamente per richieste di inference in tempo reale. Un endpoint che non viene utilizzato per più di 15 giorni è considerato sottoutilizzato. A tutti gli endpoint vengono addebitati costi in base sia al throughput impostato che al periodo di tempo in cui l'endpoint è attivo. [Per l'endpoint non utilizzato negli ultimi 15 giorni, si consiglia di definire una politica di scalabilità per la risorsa utilizzando Application Autoscaling](#). Per un endpoint che non è stato utilizzato negli ultimi 30 giorni e che dispone di una politica di scalabilità automatica definita, consigliamo di utilizzare l'inferenza asincrona o di eliminarlo. I risultati di questi controlli vengono aggiornati automaticamente una volta al giorno e possono essere visualizzati nella categoria sulla console. CostOptimization Trusted Advisor

Per visualizzare lo stato di utilizzo di tutti gli endpoint e i consigli corrispondenti

1. Accedi Console di gestione AWS e apri la Trusted Advisor console.
2. Nel riquadro di navigazione, scegli la categoria di CostOptimization controllo.
3. Nella pagina della categoria, puoi visualizzare il riepilogo per ogni categoria di controllo:
 - Azione consigliata (rosso): Trusted Advisor consiglia un'azione per il controllo.
 - Indagine suggerita (giallo) – Trusted Advisor rileva un possibile problema per il controllo.

- Nessun problema rilevato (verde): Trusted Advisor non rileva alcun problema per il controllo.
 - Elementi esclusi (grigio): il numero di controlli che hanno elementi esclusi, ad esempio le risorse che si desidera vengano ignorate da un controllo.
4. Scegli il controllo Amazon Comprehend Underutilized Endpoints per visualizzare la descrizione del controllo e i seguenti dettagli:
- Criteri di avviso – Descrive la soglia oltre cui un controllo cambierà lo stato.
 - Operazione consigliata – Descrive le azioni consigliate per questo controllo.
 - Tabella delle risorse: una tabella che elenca i dettagli degli endpoint e lo stato di ciascuno in base ai tuoi consigli.
5. Nella tabella Risorse, se un endpoint è contrassegnato come Investigation Recommended a causa dell'avviso Non utilizzato negli ultimi 30 giorni, puoi accedere alla pagina Dettagli dell'endpoint sulla console Amazon Comprehend.
- Se non desideri più utilizzare questo endpoint, scegli Elimina.
 - Selezionare di nuovo Delete (Elimina) per confermare l'eliminazione. Viene visualizzata la pagina dei dettagli del modello personalizzato. Verificate che accanto all'endpoint eliminato sia indicata la dicitura «Eliminazione». Una volta eliminato, l'endpoint viene rimosso dall'elenco degli endpoint.
6. Nella tabella Risorse sulla Trusted Advisor console, se un endpoint è contrassegnato con lo stato Investigation Recommended perché non è stato utilizzato negli ultimi 15 giorni e se è stato AutoScaling disabilitato, puoi accedere alla pagina dei dettagli dell'endpoint sulla console Amazon Comprehend per modificare l'endpoint.
- Se desideri ridurre il throughput configurato per questo endpoint, fai clic su Modifica. Inserisci il numero aggiornato di unità di inferenza da assegnare all'endpoint, quindi seleziona la casella di controllo per confermare e quindi scegli Modifica endpoint. Una volta completato l'aggiornamento, lo stato verrà visualizzato come Pronto.
 - Se desideri impostare automaticamente il provisioning degli endpoint sull'endpoint anziché regolare manualmente la configurazione del throughput, ti consigliamo di utilizzare Application Autoscaling.
7. Nella tabella delle risorse sulla Trusted Advisor console, se un endpoint è contrassegnato con lo stato Nessun problema rilevato a causa del motivo Utilizzato attivamente, significa che l'endpoint viene utilizzato attivamente per eseguire richieste di inferenza in tempo reale e non è consigliata alcuna azione.

Ecco un esempio che mostra la visualizzazione delle categorie sulla CostOptimization console: Trusted Advisor



Rischio di accesso agli endpoint di Amazon Comprehend

Il controllo del rischio di accesso agli endpoint di Amazon Comprehend valuta le AWS Key Management Service (AWS KMS) autorizzazioni chiave per un endpoint in cui il modello sottostante è stato crittografato utilizzando chiavi gestite dal cliente. Se la chiave gestita dal cliente è disabilitata o la politica delle chiavi è stata modificata per modificare le autorizzazioni consentite per Amazon Comprehend, la disponibilità degli endpoint potrebbe risentirne. Se la chiave è stata disabilitata, ti consigliamo di abilitarla. Se la politica chiave è stata modificata e desideri continuare a utilizzare questo endpoint, ti consigliamo di aggiornare la politica chiave. I risultati del controllo vengono aggiornati automaticamente più volte durante il giorno. Questo controllo può essere visualizzato nella categoria Fault Tolerance della Trusted Advisor console.

Per visualizzare lo stato AWS KMS chiave degli endpoint Amazon Comprehend

1. Accedi a Console di gestione AWS e apri la Trusted Advisor console.
2. Nel riquadro di navigazione, scegli la categoria di FaultTolerancecontrollo.
3. Nella pagina della categoria, puoi visualizzare il riepilogo per ogni categoria di controllo:

- Azione consigliata (rosso): Trusted Advisor consiglia un'azione per il controllo.
 - Indagine consigliata (giallo): Trusted Advisor rileva un possibile problema durante il controllo.
 - Nessun problema rilevato (verde): Trusted Advisor non rileva alcun problema durante il controllo.
 - Elementi esclusi (grigio) – Il numero di controlli che hanno escluso degli elementi, come ad esempio delle risorse che non si desidera sottoporre a un controllo.
4. Scegli Amazon Comprehend Endpoint Access Risk Check e potrai visualizzare la descrizione del controllo e i seguenti dettagli:
- Criteri di avviso: descrive la soglia alla quale un controllo cambierà lo stato.
 - Operazione consigliata – Descrive le azioni consigliate per questo controllo.
 - Tabella delle risorse: una tabella che elenca i dettagli degli endpoint crittografati KMS e lo stato di ciascuno di essi in base alle azioni consigliate.
5. Nella tabella Risorse, se un endpoint è contrassegnato con lo stato Azione consigliata, seleziona il link nella KeyId colonna KMS e verrai reindirizzato alla pagina chiave corrispondente. AWS KMS
- Per abilitare una AWS KMS chiave disabilitata, scegli Azioni chiave e seleziona Abilita.
 - Se lo stato della chiave è indicato come Abilitato, aggiorna la politica chiave scegliendo Passa alla visualizzazione dei criteri nella sezione Politica chiave. Modifica il documento della policy chiave per fornire le autorizzazioni necessarie ad Amazon Comprehend, quindi seleziona Salva modifiche.

Ecco un esempio della visualizzazione delle FaultTolerance categorie sulla Trusted Advisor console:

Fault tolerance checks

⊗ 0 Info ⚠ 0 Info ✓ 1 Info ⊖ 0 Info

Action recommended Investigation recommended No problems detected Excluded items

Filter by tag [Learn more about using tags](#)

Tag Key Tag Value Reset Apply filter

View by: All checks

- ▶ ✓ **AWS Lambda VPC-enabled Functions without Multi-AZ Redundancy** Refreshed: 11 hours ago
Checks for VPC-enabled Lambda functions that are vulnerable to service interruption in a single availability zone.
- ▶ ⊖ **Amazon Aurora DB Instance Accessibility**
Checks for cases where an Amazon Aurora DB cluster has both private and public instances.
- ▶ ⊖ **Amazon EBS Snapshots**
Checks the age of the snapshots for your Amazon Elastic Block Store (Amazon EBS) volumes (available or in-use).
- ▶ ⊖ **Amazon EC2 Availability Zone Balance**
Checks the distribution of Amazon Elastic Compute Cloud (Amazon EC2) instances across Availability Zones in a region.

Questi controlli e i relativi risultati possono essere visualizzati anche facendo riferimento alla Trusted Advisor sezione dell' Supporto AWS API.

Per ulteriori informazioni sulla configurazione degli allarmi utilizzando CloudWatch, vedi: [Creazione di Trusted Advisor allarmi](#) utilizzando CloudWatch. Per una serie completa di controlli sulle Trusted Advisor migliori pratiche, consulta: lista di controllo delle [AWS Trusted Advisor migliori pratiche](#).

Eliminazione degli endpoint Amazon Comprehend

Una volta che l'endpoint non è più necessario, è consigliabile eliminarlo in modo da non incorrere in costi aggiuntivi. Puoi creare facilmente un altro endpoint ogni volta che ne hai bisogno dalla sezione Endpoints.

Per eliminare un endpoint (console)

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Dal menu a sinistra, scegli Endpoints.

3. Nella tabella Endpoints individua l'endpoint che desideri eliminare. Puoi cercare o filtrare tutti gli endpoint per trovare quello che ti serve.
4. Seleziona la casella di controllo dell'endpoint per l'endpoint che desideri eliminare. In alto a destra nella tabella degli endpoint, seleziona l'icona Azioni.
5. Scegli Elimina.
6. Selezionare di nuovo Delete (Elimina) per confermare l'eliminazione. Viene visualizzata la pagina degli endpoint. Verificate che accanto all'endpoint eliminato sia presente la dicitura Eliminazione. Quando viene eliminato, l'endpoint viene rimosso dall'elenco degli endpoint.

Per eliminare un endpoint ()AWS CLI

L'esempio seguente dimostra l'utilizzo dell>DeleteEndpointoperazione con la AWS CLI.

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:region:account-id endpoint/endpoint name
```

Se l'azione ha esito positivo, Amazon Comprehend risponde con una risposta HTTP 200 con un corpo HTTP vuoto.

Scalabilità automatica con endpoint

Invece di regolare manualmente il numero di unità di inferenza fornite per gli endpoint di classificazione dei documenti e gli endpoint di riconoscimento delle entità, puoi utilizzare la scalabilità automatica per impostare automaticamente il provisioning degli endpoint in base alle tue esigenze di capacità.

Esistono due modi per utilizzare la scalabilità automatica per regolare il numero di unità di inferenza fornite per l'endpoint:

- [Monitoraggio degli obiettivi](#): imposta la scalabilità automatica per adattare il provisioning degli endpoint alle esigenze di capacità in base all'utilizzo.
- [Dimensionamento pianificato](#): imposta la scalabilità automatica per adattare il provisioning degli endpoint alle esigenze di capacità secondo una pianificazione specifica.

È possibile impostare il ridimensionamento automatico solo con AWS Command Line Interface (AWS CLI). Per ulteriori informazioni sull'auto scaling, consulta [What is Application Auto Scaling?](#)

Monitoraggio degli obiettivi

Con il monitoraggio degli obiettivi, puoi adattare il provisioning degli endpoint alle tue esigenze di capacità in base all'utilizzo. Il numero di unità di inferenza si regola automaticamente in modo che la capacità utilizzata rientri nella percentuale target della capacità fornita. È possibile utilizzare il monitoraggio degli obiettivi per far fronte a picchi temporanei di utilizzo degli endpoint di classificazione dei documenti e degli endpoint di riconoscimento delle entità. Per ulteriori informazioni, consulta [Policy di dimensionamento con monitoraggio degli obiettivi per Application Auto Scaling](#).

Note

I seguenti esempi sono formattati per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Configurazione del tracciamento degli obiettivi

Per configurare il tracciamento delle destinazioni per un endpoint, si utilizzano AWS i comandi CLI per registrare un target scalabile e quindi creare una politica di scalabilità. Il target scalabile definisce le unità di inferenza come risorsa utilizzata per regolare il provisioning degli endpoint e la policy di scalabilità definisce le metriche che controllano la scalabilità automatica della capacità fornita.

Per impostare il tracciamento degli obiettivi

1. Registrazione di una destinazione dimensionabile. Gli esempi seguenti registrano un target scalabile per regolare il provisioning degli endpoint con una capacità minima di 1 unità di inferenza e una capacità massima di 2 unità di inferenza.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando AWS CLI:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace comprehend \  
  --resource-id arn:aws:comprehend:region:account-id:document-classifier-  
endpoint/name \  
  --min-inference-units 1 --max-inference-units 2
```

```
--scalable-dimension comprehend:document-classifier-
endpoint:DesiredInferenceUnits \
--min-capacity 1 \
--max-capacity 2
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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. Per verificare la registrazione del target scalabile, utilizza il seguente comando AWS CLI:

```
aws application-autoscaling describe-scalable-targets \
--service-namespace comprehend \
--resource-id endpoint ARN
```

3. Crea una configurazione di tracciamento della destinazione per la politica di ridimensionamento e salva la configurazione in un file chiamato `config.json`. Di seguito è riportato un esempio di configurazione di tracciamento degli obiettivi per un endpoint di classificazione dei documenti che mira a mantenere la `InferenceUtilization` metrica al 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"
  }
}

```

Di seguito è riportato un esempio di endpoint di riconoscimento delle entità:

```

{
  "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. Crea una politica di scalabilità. Gli esempi seguenti creano una politica di scalabilità basata sulla configurazione di tracciamento del target definita nel config.json file.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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

```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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
```

Considerazioni

Le seguenti considerazioni si applicano quando si utilizza il tracciamento del target con gli endpoint Comprehend:

- Le metriche degli endpoint vengono emesse solo per le richieste riuscite. Le metriche non verranno visualizzate per le richieste limitate o non riuscite a causa di un errore interno del server o di un errore del cliente.
- Quando mancano i punti dati, lo stato di CloudWatch allarme di backup cambierà in. `INSUFFICIENT_DATA` Quando ciò accade, Application Auto Scaling non è in grado di scalare l'endpoint.
- La matematica metrica può essere utile per aggirare questa limitazione. Ad esempio, per utilizzare il valore 0 quando non viene riportata alcuna metrica, utilizzate la `FILL(m1, 0)` funzione where `m1` is the metric. È importante testare la configurazione per assicurarsi che si comporti come previsto. Per ulteriori opzioni, consulta [Creazione di politiche di tracciamento degli obiettivi utilizzando la matematica delle metriche](#).

Rimuovere il tracciamento degli obiettivi

Per rimuovere il tracciamento delle destinazioni per un endpoint, si utilizzano AWS i comandi CLI per eliminare la politica di scalabilità e quindi annullare la registrazione della destinazione scalabile.

Per rimuovere il tracciamento degli obiettivi

1. Elimina la politica di scalabilità. Gli esempi seguenti eliminano una politica di ridimensionamento specificata.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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 \
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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. Annullare la registrazione di una destinazione scalabile. I seguenti esempi annullano la registrazione di un target scalabile specificato.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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
```

Dimensionamento pianificato

Con la scalabilità pianificata, puoi adattare il provisioning degli endpoint alle tue esigenze di capacità secondo una pianificazione specifica. La scalabilità programmata regola automaticamente il numero di unità di inferenza per far fronte ai picchi di utilizzo in momenti specifici. È possibile utilizzare la scalabilità pianificata per gli endpoint di classificazione dei documenti e gli endpoint di riconoscimento delle entità. Per ulteriori informazioni sulla scalabilità pianificata, vedere Scalabilità [pianificata per Application Auto Scaling](#).

Note

I seguenti esempi sono formattati per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Configurazione del ridimensionamento pianificato

Per configurare la scalabilità pianificata per un endpoint, si utilizzano i comandi AWS CLI per registrare un target scalabile e quindi creare un'azione pianificata. L'obiettivo scalabile definisce le unità di inferenza come risorsa utilizzata per regolare il provisioning degli endpoint e l'azione pianificata controlla la scalabilità automatica della capacità fornita in momenti specifici.

Per impostare la scalabilità pianificata

1. Registrazione di una destinazione dimensionabile. Gli esempi seguenti registrano un obiettivo scalabile per regolare il provisioning degli endpoint con una capacità minima di 1 unità di inferenza e una capacità massima di 2 unità di inferenza.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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
```


Per un endpoint di riconoscimento delle entità, usa il seguente comando 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. Crea un'azione pianificata. Gli esempi seguenti creano un'azione pianificata per regolare automaticamente la capacità fornita ogni giorno alle 12:00 UTC con un minimo di 2 unità di inferenza e un massimo di 5 unità di inferenza. [Per ulteriori informazioni sulle espressioni cronologiche e sul ridimensionamento pianificato, vedere Schedule expression.](#)

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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
```

Rimozione del ridimensionamento pianificato

Per rimuovere il ridimensionamento pianificato per un endpoint, si utilizzano i comandi AWS CLI per eliminare l'azione pianificata e quindi annullare la registrazione dell'obiettivo scalabile.

Per rimuovere il ridimensionamento pianificato

1. Eliminare l'azione pianificata. I seguenti esempi eliminano un'azione pianificata specificata.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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. Annullare la registrazione di una destinazione scalabile. I seguenti esempi annullano la registrazione di un target scalabile specificato.

Per un endpoint di classificazione dei documenti, utilizzate il seguente comando 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
```

Per un endpoint di riconoscimento delle entità, usa il seguente comando 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
```

Tagging delle risorse

Un tag è una coppia chiave-valore che puoi aggiungere a una risorsa Amazon Comprehend come metadati. Puoi utilizzare i tag nei job di analisi, nei modelli di classificazione personalizzati, nei modelli di riconoscimento delle entità personalizzati e negli endpoint. I tag hanno due funzioni principali: organizzare le risorse e fornire un controllo degli accessi basato sui tag.

Per organizzare le risorse con i tag, puoi aggiungere la chiave di tag «Dipartimento» e i valori dei tag «Vendite» o «Legale». Puoi quindi cercare e filtrare le risorse pertinenti all'ufficio legale della tua azienda.

Per fornire un controllo degli accessi basato sui tag, crea policy IAM con autorizzazioni basate sui tag. Una policy può consentire o impedire un'operazione in base ai tag forniti nella richiesta (request-tags) o ai tag associati alla risorsa che stai chiamando (resource-tags). Per ulteriori informazioni sull'utilizzo dei tag con IAM, consulta [Controlling access using tags](#) nella IAM User Guide.

Considerazioni sull'utilizzo dei tag con Amazon Comprehend:

- Puoi aggiungere fino a 50 tag per risorsa e i tag possono essere aggiunti al momento della creazione della risorsa o retroattivamente.
- Una chiave di tag è un campo obbligatorio, ma un valore di tag è facoltativo.
- I tag non devono essere univoci tra le risorse, ma una determinata risorsa non può avere chiavi di tag duplicate.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Una chiave tag può contenere un massimo di 127 caratteri; un valore di tag può avere un massimo di 255 caratteri.
- Il prefisso `aws :` è riservato AWS all'uso; non è possibile aggiungere, modificare o eliminare tag la cui chiave inizia con `aws :`. Questi tag non vengono conteggiati ai fini del tags-per-resource limite di 50.

Note

Se prevedi di utilizzare lo schema di tagging su più AWS servizi e risorse, ricorda che altri servizi potrebbero avere requisiti diversi per i caratteri consentiti.

Argomenti

- [Taggare una nuova risorsa](#)
- [Visualizzazione, modifica ed eliminazione dei tag associati a una risorsa](#)

Taggare una nuova risorsa

Puoi aggiungere tag a un processo di analisi, un modello di classificazione personalizzato, un modello di riconoscimento delle entità personalizzato o agli endpoint.

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Seleziona la risorsa (Job di analisi, Classificazione personalizzata o Riconoscimento personalizzato delle entità) che desideri creare dal riquadro di navigazione a sinistra.
3. Fai clic su Crea lavoro (o Crea nuovo modello). Verrai indirizzato alla pagina principale di «creazione» della tua risorsa. Nella parte inferiore di questa pagina, vedrai il pannello «Tag - opzionale».

▼ **Tags - optional** [Info](#)

A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value.

Key	Value - optional	
Enter key	Enter value	Remove tag
Add tag		

Inserisci una chiave di tag e un valore di tag opzionale. Scegli Aggiungi tag per aggiungere un altro tag alla risorsa. Ripeti questo processo fino all'aggiunta di tutti i tag. Tieni presente che le chiavi dei tag devono essere uniche per risorsa.

4. Seleziona il pulsante Crea o Crea lavoro per continuare a creare la tua risorsa.

Puoi anche aggiungere tag utilizzando la AWS CLI. Questo esempio mostra come aggiungere tag con il [start-entities-detection-job](#) comando.

```
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\"}]"
```

Visualizzazione, modifica ed eliminazione dei tag associati a una risorsa

È possibile visualizzare i tag associati a un processo di analisi, un modello di classificazione personalizzato o un modello di riconoscimento delle entità personalizzato.

1. Accedi Console di gestione AWS e apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
2. Seleziona la risorsa (Analysis job, Custom classification o Custom Entity Recognition) che contiene il file con i tag che desideri visualizzare, modificare o eliminare. Viene visualizzato l'elenco dei file esistenti per la risorsa selezionata.

The screenshot shows the Amazon Comprehend console interface. On the left, there's a sidebar with navigation links: 'Real-time analysis', 'Analysis jobs' (highlighted), 'Customization' (with sub-links for 'Custom classification', 'Custom entity recognition', and 'Endpoints'), and 'Amazon Comprehend Medical' (with sub-links for 'Real-time analysis' and 'Analysis jobs'). The main content area is titled 'Analysis jobs' and includes a description: 'Analyze documents stored in Amazon S3 to find entities like events, phrases, primary language, sentiment, or personally identifiable information (PII)'. Below this, there's a section for 'Analysis jobs (1)' with a search bar, a status filter set to 'All', and a table of jobs. The table has columns for 'Name', 'Analysis type', 'Start', and 'End'. One job is listed: 'my-comprehend-analysis-job' with an analysis type of 'Key phrases', starting on 10/22/2021 at 10:43:57 AM and ending at 10:52:07 AM. Action buttons like 'Stop', 'Duplicate', and 'Create job' are visible at the top right of the job list.

3. Fai clic sul nome del file (o modello) di cui desideri visualizzare, modificare o eliminare i tag. Verrà visualizzata la pagina dei dettagli del file (o modello). Scorri verso il basso fino a visualizzare la casella Tag. Qui puoi vedere tutti i tag associati al file (o modello) selezionato.

The screenshot shows the 'Tags' section of the console. It features a table with two columns: 'Key' and 'Value'. There are two rows of tags: one with 'color' as the key and 'orange' as the value, and another with 'type' as the key and 'PDF' as the value. A 'Manage tags' button is located in the top right corner of the tag list area.

Seleziona Gestisci tag per modificare o rimuovere i tag dalla tua risorsa.

4. Fai clic sul testo che desideri modificare, quindi modifica il tag. Puoi anche rimuovere il tag selezionando Rimuovi tag. Per aggiungere un nuovo tag, seleziona Aggiungi tag, quindi inserisci il testo desiderato nei campi vuoti.

Manage my-comprehend-analysis-job - No Version Name tags

Tags [Info](#)
A tag is a label that you can add to a resource as metadata to help you organize, search, or filter your data. Each tag consists of a key and an optional value.

Key	Value - optional	
color	orange	<button>Remove tag</button>
type	PDF	<button>Remove tag</button>
<button>Add tag</button>		

CancelSave

Quando hai finito di modificare i tag, seleziona Salva.

Esempi di codice per l'utilizzo di Amazon Comprehend AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Comprehend con un kit di sviluppo AWS software (SDK).

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica chiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- [Esempi di base per l'utilizzo di Amazon Comprehend AWS SDKs](#)
 - [Azioni per l'utilizzo di Amazon Comprehend AWS SDKs](#)
 - [Utilizzo CreateDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeDocumentClassificationJob con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeTopicsDetectionJob con un AWS SDK o una CLI](#)
 - [Utilizzo DetectDominantLanguage con un AWS SDK o una CLI](#)
 - [Utilizzo DetectEntities con un AWS SDK o una CLI](#)
 - [Utilizzo DetectKeyPhrases con un AWS SDK o una CLI](#)
 - [Utilizzo DetectPiiEntities con un AWS SDK o una CLI](#)
 - [Utilizzo DetectSentiment con un AWS SDK o una CLI](#)
 - [Utilizzo DetectSyntax con un AWS SDK o una CLI](#)
 - [Utilizzo ListDocumentClassificationJobs con un AWS SDK o una CLI](#)
 - [Utilizzo ListDocumentClassifiers con un AWS SDK o una CLI](#)
 - [Utilizzo ListTopicsDetectionJobs con un AWS SDK o una CLI](#)

- [Utilizzo StartDocumentClassificationJob con un AWS SDK o una CLI](#)
- [Utilizzo StartTopicsDetectionJob con un AWS SDK o una CLI](#)
- [Scenari per l'utilizzo di Amazon Comprehend AWS SDKs](#)
 - [Creazione di un'app in streaming Amazon Transcribe](#)
 - [Creare un chatbot Amazon Lex per coinvolgere i visitatori del sito web](#)
 - [Creare un'applicazione web che invia e recupera i messaggi utilizzando Amazon SQS](#)
 - [Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio](#)
 - [Rileva gli elementi del documento con Amazon Comprehend e un SDK AWS](#)
 - [Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS](#)
 - [Esegui un processo di modellazione tematica di Amazon Comprehend su dati di esempio utilizzando un SDK AWS](#)
 - [Addestra un classificatore Amazon Comprehend personalizzato e classifica i documenti utilizzando un SDK AWS](#)

Esempi di base per l'utilizzo di Amazon Comprehend AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon AWS SDKs Comprehend con.

Esempi

- [Azioni per l'utilizzo di Amazon Comprehend AWS SDKs](#)
 - [Utilizzo CreateDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeDocumentClassificationJob con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeDocumentClassifier con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeTopicsDetectionJob con un AWS SDK o una CLI](#)
 - [Utilizzo DetectDominantLanguage con un AWS SDK o una CLI](#)
 - [Utilizzo DetectEntities con un AWS SDK o una CLI](#)
 - [Utilizzo DetectKeyPhrases con un AWS SDK o una CLI](#)
 - [Utilizzo DetectPiiEntities con un AWS SDK o una CLI](#)
 - [Utilizzo DetectSentiment con un AWS SDK o una CLI](#)
 - [Utilizzo DetectSyntax con un AWS SDK o una CLI](#)

- [Utilizzo ListDocumentClassificationJobs con un AWS SDK o una CLI](#)
- [Utilizzo ListDocumentClassifiers con un AWS SDK o una CLI](#)
- [Utilizzo ListTopicsDetectionJobs con un AWS SDK o una CLI](#)
- [Utilizzo StartDocumentClassificationJob con un AWS SDK o una CLI](#)
- [Utilizzo StartTopicsDetectionJob con un AWS SDK o una CLI](#)

Azioni per l'utilizzo di Amazon Comprehend AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni Amazon Comprehend con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per configurare ed eseguire il codice.

Questi estratti chiamano l'API Amazon Comprehend e sono estratti di codice di programmi più grandi che devono essere eseguiti in modo contestuale. È possibile visualizzare le azioni nel contesto in [Scenari per l'utilizzo di Amazon Comprehend AWS SDKs](#).

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta la [documentazione di riferimento dell'API Amazon Comprehend](#).

Esempi

- [Utilizzo CreateDocumentClassifier con un AWS SDK o una CLI](#)
- [Utilizzo DeleteDocumentClassifier con un AWS SDK o una CLI](#)
- [Utilizzo DescribeDocumentClassificationJob con un AWS SDK o una CLI](#)
- [Utilizzo DescribeDocumentClassifier con un AWS SDK o una CLI](#)
- [Utilizzo DescribeTopicsDetectionJob con un AWS SDK o una CLI](#)
- [Utilizzo DetectDominantLanguage con un AWS SDK o una CLI](#)
- [Utilizzo DetectEntities con un AWS SDK o una CLI](#)
- [Utilizzo DetectKeyPhrases con un AWS SDK o una CLI](#)
- [Utilizzo DetectPiiEntities con un AWS SDK o una CLI](#)
- [Utilizzo DetectSentiment con un AWS SDK o una CLI](#)
- [Utilizzo DetectSyntax con un AWS SDK o una CLI](#)
- [Utilizzo ListDocumentClassificationJobs con un AWS SDK o una CLI](#)
- [Utilizzo ListDocumentClassifiers con un AWS SDK o una CLI](#)
- [Utilizzo ListTopicsDetectionJobs con un AWS SDK o una CLI](#)

- [Utilizzo StartDocumentClassificationJob con un AWS SDK o una CLI](#)
- [Utilizzo StartTopicsDetectionJob con un AWS SDK o una CLI](#)

Utilizzo **CreateDocumentClassifier** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateDocumentClassifier.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come creare un classificatore di documenti per classificare i documenti

L'esempio create-document-classifier seguente inizia il processo di addestramento per un modello di classificatore di documenti. Il file dei dati di addestramento training.csv si trova nel tag --input-data-config. training.csv è un documento a due colonne in cui le etichette o le classificazioni sono fornite nella prima colonna e i documenti nella seconda colonna.

```
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
```

Output:

```
{
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier"
}
```

Per ulteriori informazioni, consulta [Classificazione personalizzata](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [CreateDocumentClassifier AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
    }
}
}

```

- Per i dettagli sull'API, consulta la [CreateDocumentClassifier](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [CreateDocumentClassifier AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

TRY.

```

oo_result = lo_cpd->createdocumentclassifier(
    iv_documentclassifiername = iv_classifier_name
    iv_languagecode = iv_language_code
    io_inputdataconfig = NEW /aws1/cl_cpddocclifierinpdatt00(
        iv_s3uri = iv_training_s3_uri
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_mode = iv_mode
).
MESSAGE 'Document classifier creation started.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdresrclimitexcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdtomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdtomanytagsex.

```



```
MESSAGE 'Too many tags.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [CreateDocumentClassifier](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteDocumentClassifier** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteDocumentClassifier`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come eliminare un classificatore di documenti personalizzato

L'esempio `delete-document-classifier` seguente elimina un modello di classificatore di documenti personalizzato.

```
aws comprehend delete-document-classifier \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestione degli endpoint di Amazon Comprehend](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DeleteDocumentClassifier AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [DeleteDocumentClassifier AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->deleteddocumentclassifier(  
        iv_documentclassifierarn = iv_classifier_arn  
    ).  
    MESSAGE 'Document classifier deleted.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdresourcefoundex.  
    MESSAGE 'Resource not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdresourceinuseex.  
    MESSAGE 'Resource in use.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DeleteDocumentClassifier](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeDocumentClassificationJob** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeDocumentClassificationJob`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come descrivere un processo di classificazione dei documenti

L'esempio `describe-document-classification-job` seguente ottiene le proprietà di un processo asincrono di classificazione di documenti.

```
aws comprehend describe-document-classification-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Classificazione personalizzata](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DescribeDocumentClassificationJob AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [DescribeDocumentClassificationJob AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->describedocclassificationjob(
        iv_jobid = iv_job_id
    ).
    MESSAGE 'Document classification job described.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdjobnotfoundex.
    MESSAGE 'Job not found.' TYPE 'E'.
CATCH /aws1/cx_cpdtomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [DescribeDocumentClassificationJob](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeDocumentClassifier** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeDocumentClassifier`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come descrivere un classificatore di documenti

L'esempio `describe-document-classifier` seguente ottiene le proprietà di un modello di classificatore di documenti personalizzato.

```
aws comprehend describe-document-classifier \
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
  classifier/example-classifier-1
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DescribeDocumentClassifier AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [DescribeDocumentClassifier AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->describedocumentclassifier(
        iv_documentclassifierarn = iv_classifier_arn
    ).
    MESSAGE 'Document classifier described.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdrourcenotfoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cpdsinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [DescribeDocumentClassifier](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeTopicsDetectionJob** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeTopicsDetectionJob`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Eseguire un processo di modellazione di argomenti in base a dati di esempio](#)

CLI

AWS CLI

Come descrivere un processo di rilevamento di argomenti

L'esempio `describe-topics-detection-job` seguente ottiene le proprietà di un processo asincrono di rilevamento degli argomenti.

```
aws comprehend describe-topics-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Analisi asincrona degli approfondimenti di Amazon Comprehend](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DescribeTopicsDetectionJob AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [DescribeTopicsDetectionJob AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->describetopicdetectionjob(  
        iv_jobid = iv_job_id  
    ).  
    MESSAGE 'Topics detection job described.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdjobnotfoundex.  
    MESSAGE 'Job not found.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DescribeTopicsDetectionJob](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectDominantLanguage** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectDominantLanguage.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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");
    }
}
```

- Per i dettagli sull'API, consulta la [DetectDominantLanguage](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare la lingua dominante del testo di input

Il comando `detect-dominant-language` seguente analizza il testo di input e identifica la lingua dominante. Viene inoltre restituito il punteggio di attendibilità del modello pre-addestrato.

```
aws comprehend detect-dominant-language \
  --text "It is a beautiful day in Seattle."
```

Output:

```
{
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9877256155014038
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lingua dominante](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DetectDominantLanguage AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
        }
    }
}

```

- Per i dettagli sull'API, consulta la [DetectDominantLanguage](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [DetectDominantLanguage AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

TRY.

```
oo_result = lo_cpd->detectdominantlanguage( iv_text = iv_text ).
```

```
MESSAGE 'Languages detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextrsizeintexcdex.  
MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdtinternalserverex.  
MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdtinvalidrequestex.  
MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectDominantLanguage](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectEntities** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectEntities.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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");
    }
}
```

- Per i dettagli sull'API, consulta la [DetectEntities](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare entità denominate nel testo di input

L'esempio detect-entities seguente analizza il testo di input e restituisce le entità denominate. Per ogni previsione viene inoltre restituito il punteggio di attendibilità del modello pre-addestrato.

```
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."
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Entità](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DetectEntities AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
    }
}
}

```

- Per i dettagli sull'API, consulta la [DetectEntities](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [DetectEntities AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->detectentities(
        iv_text = iv_text
        iv_languagecode = iv_language_code
    ).
    MESSAGE 'Entities detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextsizelmtexcdex.
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdunsuppedlanguageex.
    MESSAGE 'Unsupported language.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_cpdinvaliddrequestex.

```

```
MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectEntities](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectKeyPhrases** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectKeyPhrases.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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");
    }
}

```

- Per i dettagli sull'API, consulta la [DetectKeyPhrases](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare le frasi chiave nel testo di input

L'esempio detect-key-phrases seguente analizza il testo di input e identifica le frasi nominali chiave. Per ogni previsione viene inoltre restituito il punteggio di attendibilità del modello pre-addestrato.

```
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."
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Frase chiave](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DetectKeyPhrases AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
        }
    }
}

```

- Per i dettagli sull'API, consulta la [DetectKeyPhrases](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [DetectKeyPhrases AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->detectkeyphrases(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Key phrases detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextsize limitexcdex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdunsupp languageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternal serverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalid requestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectKeyPhrases](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectPiiEntities** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectPiiEntities.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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 identifiable 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}");
            }
        }
    }
}

```

- Per i dettagli sull'API, consulta la [DetectPiiEntities](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare entità PII nel testo di input

L'esempio `detect-pii-entities` seguente analizza il testo di input e identifica le entità che contengono informazioni di identificazione personale (PII). Per ogni previsione viene inoltre restituito il punteggio di attendibilità del modello pre-addestrato.

```

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."

```

Output:

```

{
  "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
    }
]
}

```

Per ulteriori informazioni, consulta [Informazioni di identificazione personale \(PII\)](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DetectPiiEntities AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [DetectPiiEntities AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->detectpiientities(
        iv_text = iv_text
        iv_languagecode = iv_language_code
    ).
    MESSAGE 'PII entities detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextsizelmtexcdex.
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdunsuppedlanguageex.
    MESSAGE 'Unsupported language.' TYPE 'E'.

```



```
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectPiiEntities](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectSentiment** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectSentiment.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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 DetectSentimentAsync method to analyze the
        /// supplied text and determine the overall 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");
        }
    }
}

```

- Per i dettagli sull'API, consulta la [DetectSentiment](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare il sentiment di un testo di input

L'esempio detect-sentiment seguente analizza il testo di input e restituisce un'inferenza del sentiment prevalente (POSITIVE, NEUTRAL, MIXED o NEGATIVE).

```
aws comprehend detect-sentiment \
```

```
--language-code en \  
--text "It is a beautiful day in Seattle"
```

Output:

```
{  
  "Sentiment": "POSITIVE",  
  "SentimentScore": {  
    "Positive": 0.9976957440376282,  
    "Negative": 9.653854067437351e-05,  
    "Neutral": 0.002169104292988777,  
    "Mixed": 3.857641786453314e-05  
  }  
}
```

Per ulteriori informazioni, consulta [Sentiment](#) nella Guida per gli sviluppatori di Amazon Comprehend

- Per i dettagli sull'API, consulta [DetectSentiment AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
        }
    }
}
```

- Per i dettagli sull'API, consulta la [DetectSentiment](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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:
            pass
```

```
return response
```

- Per i dettagli sull'API, consulta [DetectSentiment AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->detectsentiment(  
        iv_text = iv_text  
        iv_languagecode = iv_language_code  
    ).  
    MESSAGE 'Sentiment detected.' TYPE 'I'.  
CATCH /aws1/cx_cpdtextsize limitex.  
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.  
CATCH /aws1/cx_cpdunsupp languageex.  
    MESSAGE 'Unsupported language.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternal serverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalid requestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectSentiment](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DetectSyntax** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DetectSyntax.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Rilevare gli elementi del documento](#)

.NET

SDK per .NET

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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");
    }
}

```

- Per i dettagli sull'API, consulta la [DetectSyntax](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come rilevare le parti del discorso in un testo di input

L'esempio detect-syntax seguente analizza la sintassi del testo di input e restituisce le diverse parti del discorso. Per ogni previsione viene inoltre restituito il punteggio di attendibilità del modello pre-addestrato.

```

aws comprehend detect-syntax \
  --language-code en \
  --text "It is a beautiful day in Seattle."

```

Output:

```

{
  "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
        }
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Analisi della sintassi](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [DetectSyntax AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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);
    }
}
}

```

- Per i dettagli sull'API, consulta la [DetectSyntax](#) sezione AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [DetectSyntax AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->detectsyntax(
        iv_text = iv_text
        iv_languagecode = iv_language_code
    ).
    MESSAGE 'Syntax tokens detected.' TYPE 'I'.
CATCH /aws1/cx_cpdtextsizelmtexcdex.
    MESSAGE 'Text size exceeds limit.' TYPE 'E'.
CATCH /aws1/cx_cpdunsuppedlanguageex.
    MESSAGE 'Unsupported language.' TYPE 'E'.

```

```
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [DetectSyntax](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListDocumentClassificationJobs** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare ListDocumentClassificationJobs.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come un elenco di tutti i processi di classificazione dei documenti

L'esempio list-document-classification-jobs seguente elenca tutti i processi di classificazione dei documenti.

```
aws comprehend list-document-classification-jobs
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Classificazione personalizzata](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [ListDocumentClassificationJobs AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [ListDocumentClassificationJobs AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->listdocclassificationjobs( ).  
    MESSAGE 'Document classification jobs listed.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [ListDocumentClassificationJobs](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListDocumentClassifiers** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListDocumentClassifiers`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come un elenco di tutti i classificatori dei documenti

L'esempio `list-document-classifiers` seguente elenca tutti i modelli di classificatori di documenti addestrati e in corso di addestramento.

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

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [ListDocumentClassifiers AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [ListDocumentClassifiers AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->listdocumentclassifiers( ).  
    MESSAGE 'Document classifiers listed.' TYPE 'I'.  
CATCH /aws1/cx_cpdinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpdinvalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [ListDocumentClassifiers](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListTopicsDetectionJobs** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListTopicsDetectionJobs`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Eseguire un processo di modellazione di argomenti in base a dati di esempio](#)

CLI

AWS CLI

Come elencare tutti i processi di rilevamento degli argomenti

L'esempio `list-topics-detection-jobs` seguente elenca tutti i processi asincroni di rilevamento di argomenti in corso e completati.

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

Output:

```
{
  "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": "123456abcdeb0e11022f22a11EXAMPLE2",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:topics-
detection-job/123456abcdeb0e11022f22a11EXAMPLE2",
```

```

    "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Analisi asincrona degli approfondimenti di Amazon Comprehend](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [ListTopicsDetectionJobs AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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
```

- Per i dettagli sull'API, consulta [ListTopicsDetectionJobs AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
TRY.  
    oo_result = lo_cpd->listtopicsdetectionjobs( ).  
    MESSAGE 'Topics detection jobs listed.' TYPE 'I'.  
CATCH /aws1/cx_cpinvalidrequestex.  
    MESSAGE 'Invalid request.' TYPE 'E'.  
CATCH /aws1/cx_cpdtoomanyrequestsex.  
    MESSAGE 'Too many requests.' TYPE 'E'.  
CATCH /aws1/cx_cpinvalidfilterex.  
    MESSAGE 'Invalid filter.' TYPE 'E'.  
CATCH /aws1/cx_cpinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [ListTopicsDetectionJobs](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **StartDocumentClassificationJob** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare StartDocumentClassificationJob.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Addestrare un classificatore personalizzato e classificare i documenti](#)

CLI

AWS CLI

Come avviare un processo di classificazione dei documenti

L'esempio `start-document-classification-job` seguente avvia un processo di classificazione dei documenti con un modello personalizzato su tutti i file all'indirizzo specificato dal tag `--input-data-config`. In questo esempio, il bucket S3 di input contiene `SampleSMStext1.txt`, `SampleSMStext2.txt` e `SampleSMStext3.txt`. Il modello è stato precedentemente addestrato in base alla classificazione dei documenti dei messaggi SMS spam e non spam o "indesiderati". Al completamento del processo, `output.tar.gz` viene collocato nella posizione specificata dal tag `--output-data-config`. `output.tar.gz` contiene `predictions.jsonl`, che riporta la classificazione di ogni documento. L'output Json viene stampato su una riga per file, ma di seguito è formattato per ottimizzarne la leggibilità.

```
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
```

Contenuto di `SampleSMStext1.txt`:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Contenuto di `SampleSMStext2.txt`:

```
"Hi, when do you want me to pick you up from practice?"
```

Contenuto di `SampleSMStext3.txt`:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Output:

```
{
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-job/e758dd56b824aa717ceab551fEXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Contenuto di `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}]}
```

Per ulteriori informazioni, consulta [Classificazione personalizzata](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [StartDocumentClassificationJob AWS CLI Command Reference](#).

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [StartDocumentClassificationJob AWSSDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

TRY.

```

oo_result = lo_cpd->startdocclassificationjob(
    iv_jobname = iv_job_name
    iv_documentclassifierarn = iv_classifier_arn
    io_inputdataconfig = NEW /aws1/cl_cpdinputdataconfig(
        iv_s3uri = iv_input_s3_uri
        iv_inputformat = iv_input_format
    )
    io_outputdataconfig = NEW /aws1/cl_cpdoutputdataconfig(
        iv_s3uri = iv_output_s3_uri
    )
)

```

```

        iv_dataaccessrolelearn = iv_data_access_role_arn
    ).
    MESSAGE 'Document classification job started.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdresourceNotFoundex.
    MESSAGE 'Resource not found.' TYPE 'E'.
CATCH /aws1/cx_cpdresourceunavailablex.
    MESSAGE 'Resource unavailable.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
    MESSAGE 'KMS key validation error.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanytagsex.
    MESSAGE 'Too many tags.' TYPE 'E'.
CATCH /aws1/cx_cpdresrcLimitExcdex.
    MESSAGE 'Resource limit exceeded.' TYPE 'E'.
CATCH /aws1/cx_cpdinternalserverex.
    MESSAGE 'Internal server error occurred.' TYPE 'E'.
ENDTRY.

```

- Per i dettagli sulle API, [StartDocumentClassificationJob](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **StartTopicsDetectionJob** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare StartTopicsDetectionJob.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- [Eseguire un processo di modellazione di argomenti in base a dati di esempio](#)

.NET

SDK per .NET

Note

C'è altro su GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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 method 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}");
    }
}

```

- Per i dettagli sull'API, consulta la [StartTopicsDetectionJob](#) sezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Come avviare un processo di analisi del rilevamento degli argomenti

L'esempio `start-topics-detection-job` seguente avvia un processo asincrono di rilevamento degli argomenti per tutti i file che si trovano all'indirizzo specificato dal tag `--input-data-config`. Al completamento del processo, la cartella output viene collocata nella posizione specificata dal tag `--output-data-config`. output contiene `topic-terms.csv` e `doc-topics.csv`. Il primo file di output (`topic-terms.csv`) è un elenco di argomenti nella raccolta. Per ogni argomento, per impostazione predefinita l'elenco include i termini principali per argomento in base al loro peso. Il secondo file (`doc-topics.csv`) elenca i documenti associati a un argomento e la proporzione del documento interessato dall'argomento.

```

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
```

Output:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

Per ulteriori informazioni, consulta [Modellazione degli argomenti](#) nella Guida per gli sviluppatori di Amazon Comprehend.

- Per i dettagli sull'API, consulta [StartTopicsDetectionJob AWS CLI](#) Command Reference.

Python

SDK per Python (Boto3)

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice 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

```

- Per i dettagli sull'API, consulta [StartTopicsDetectionJob AWS SDK for Python \(Boto3\) API Reference](#).

SAP ABAP

SDK per SAP ABAP

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

TRY.
    oo_result = lo_cpd->starttopicsdetectionjob(
        iv_jobname = iv_job_name
        io_inputdataconfig = NEW /aws1/cl_cpdinputdataconfig(
            iv_s3uri = iv_input_s3_uri
            iv_inputformat = iv_input_format
        )
        io_outputdataconfig = NEW /aws1/cl_cpdoutputdataconfig(
            iv_s3uri = iv_output_s3_uri
        )
        iv_dataaccessrolelearn = iv_data_access_role_arn
    ).
    MESSAGE 'Topics detection job started.' TYPE 'I'.
CATCH /aws1/cx_cpdinvalidrequestex.
    MESSAGE 'Invalid request.' TYPE 'E'.
CATCH /aws1/cx_cpdtoomanyrequestsex.
    MESSAGE 'Too many requests.' TYPE 'E'.
CATCH /aws1/cx_cpdkmskeyvalidationex.
    MESSAGE 'KMS key validation error.' TYPE 'E'.

```

```
CATCH /aws1/cx_cpdtoomanytagsex.  
    MESSAGE 'Too many tags.' TYPE 'E'.  
CATCH /aws1/cx_cpdrsrcrlimitexcdex.  
    MESSAGE 'Resource limit exceeded.' TYPE 'E'.  
CATCH /aws1/cx_cpdinternalserverex.  
    MESSAGE 'Internal server error occurred.' TYPE 'E'.  
ENDTRY.
```

- Per i dettagli sulle API, [StartTopicsDetectionJob](#) consulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l'utilizzo di Amazon Comprehend AWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon Comprehend con AWS SDKs. Questi scenari illustrano come eseguire attività specifiche chiamando più funzioni all'interno di Amazon Comprehend o in combinazione con altri Servizi AWS. Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- [Creazione di un'app in streaming Amazon Transcribe](#)
- [Creare un chatbot Amazon Lex per coinvolgere i visitatori del sito web](#)
- [Creare un'applicazione web che invia e recupera i messaggi utilizzando Amazon SQS](#)
- [Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio](#)
- [Rileva gli elementi del documento con Amazon Comprehend e un SDK AWS](#)
- [Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS](#)
- [Esegui un processo di modellazione tematica di Amazon Comprehend su dati di esempio utilizzando un SDK AWS](#)

- [Addestra un classificatore Amazon Comprehend personalizzato e classifica i documenti utilizzando un SDK AWS](#)

Creazione di un'app in streaming Amazon Transcribe

L'esempio di codice seguente mostra come creare un'applicazione che registra, trascrive e traduce l'audio in tempo reale e invia tramite e-mail i risultati.

JavaScript

SDK per JavaScript (v3)

Mostra come utilizzare Amazon Transcribe per creare un'applicazione che registra, trascrive e traduce l'audio in tempo reale e invia i risultati per e-mail tramite Amazon Simple Email Service (Amazon SES).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. [GitHub](#)

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creare un chatbot Amazon Lex per coinvolgere i visitatori del sito web

Gli esempi di codice seguenti mostrano come creare un chatbot per coinvolgere i visitatori del sito web.

Java

SDK per Java 2.x

Mostra come utilizzare l'API Amazon Lex per creare un chatbot all'interno di un'applicazione web per coinvolgere i visitatori del sito web.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su [GitHub](#).

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

JavaScript

SDK per JavaScript (v3)

Mostra come utilizzare l'API Amazon Lex per creare un chatbot all'interno di un'applicazione web per coinvolgere i visitatori del sito web.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, consulta l'esempio completo [Costruire un chatbot Amazon Lex](#) nella guida per gli AWS SDK per JavaScript sviluppatori.

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creare un'applicazione web che invia e recupera i messaggi utilizzando Amazon SQS

Gli esempi di codice seguenti mostrano come creare un'applicazione di messaggistica utilizzando Amazon SQS.

Java

SDK per Java 2.x

Mostra come utilizzare l'API Amazon SQS per sviluppare una REST API Spring che invia e recupera i messaggi.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su [GitHub](#).

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon SQS

Kotlin

SDK per Kotlin

Mostra come utilizzare l'API Amazon SQS per sviluppare una REST API Spring che invia e recupera i messaggi.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su [GitHub](#).

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Amazon SQS

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crea un'applicazione che analizza il feedback dei clienti e sintetizza l'audio

Il seguente esempio di codice spiega come creare un'applicazione che analizza schede dei commenti dei clienti, le traduce dalla loro lingua originale, ne determina la valutazione e genera un file audio dal testo tradotto.

.NET

SDK per .NET

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in [GitHub](#).

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK per Java 2.x

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli

ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in [GitHub](#).

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK per JavaScript (v3)

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in [GitHub](#). I seguenti estratti mostrano come AWS SDK per JavaScript viene utilizzato all'interno delle funzioni 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 };
};
```

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK per Ruby

Questa applicazione di esempio analizza e archivia le schede di feedback dei clienti. In particolare, soddisfa l'esigenza di un hotel fittizio a New York City. L'hotel riceve feedback dagli ospiti in varie lingue sotto forma di schede di commento fisiche. Tale feedback viene caricato nell'app tramite un client Web. Dopo aver caricato l'immagine di una scheda di commento, vengono eseguiti i seguenti passaggi:

- Il testo viene estratto dall'immagine utilizzando Amazon Textract.
- Amazon Comprehend determina il sentiment del testo estratto e la sua lingua.
- Il testo estratto viene tradotto in inglese utilizzando Amazon Translate.
- Amazon Polly sintetizza un file audio dal testo estratto.

L'app completa può essere implementata con AWS CDK. Per il codice sorgente e le istruzioni di distribuzione, consulta il progetto in. [GitHub](#)

Servizi utilizzati in questo esempio

- Amazon Comprehend

- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva gli elementi del documento con Amazon Comprehend e un SDK AWS

L'esempio di codice seguente mostra come:

- Rilevare lingue, entità e frasi chiave in un documento.
- Rilevare le informazioni di identificazione personale (PII) in un documento.
- Rilevare il sentiment di un documento.
- Rilevare gli elementi di sintassi in un documento.

Python

SDK per Python (Boto3)

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea una classe per il wrapping delle azioni di 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
```

Chiama le funzioni della classe wrapper per rilevare entità, frasi e altri elementi in un 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)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
 - [DetectDominantLanguage](#)
 - [DetectEntities](#)
 - [DetectKeyPhrases](#)
 - [DetectPiiEntities](#)
 - [DetectSentiment](#)
 - [DetectSyntax](#)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Rileva le entità nel testo estratto da un'immagine utilizzando un SDK AWS

L'esempio di codice seguente mostra come utilizzare Amazon Comprehend per rilevare le entità nel testo estratto da Amazon Textract da un'immagine archiviata in Amazon S3.

Python

SDK per Python (Boto3)

Mostra come utilizzarlo AWS SDK per Python (Boto3) in un notebook Jupyter per rilevare entità nel testo estratto da un'immagine. In questo esempio viene utilizzato Amazon Textract per estrarre il testo da un'immagine archiviata in Amazon Simple Storage Service (Amazon S3) e Amazon Comprehend per rilevare le entità nel testo estratto.

Questo esempio è un notebook Jupyter e deve essere eseguito in un ambiente in grado di ospitare notebook. Per istruzioni su come eseguire l'esempio utilizzando Amazon SageMaker AI, consulta le istruzioni in [TextractAndComprehendNotebook.ipynb](#).

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. [GitHub](#)

Servizi utilizzati in questo esempio

- Amazon Comprehend
- Simple Storage Service (Amazon S3)
- Amazon Textract

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esegui un processo di modellazione tematica di Amazon Comprehend su dati di esempio utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Eseguire un processo di modellazione di argomenti di Amazon Comprehend in base a dati di esempio.
- Ottenere informazioni sul processo.

- Estrarre i dati di output del processo da Amazon S3.

Python

SDK per Python (Boto3)

Note

C'è altro su [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea una classe wrapper per chiamare le azioni di modellazione di argomenti di 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
            all                bucket. If multiple objects have the same prefix,
                                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

```

Utilizza la classe wrapper per eseguire un processo di modellazione di argomenti e ottenere i dati del processo.

```

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)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
 - [DescribeTopicsDetectionJob](#)
 - [ListTopicsDetectionJobs](#)
 - [StartTopicsDetectionJob](#)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di Amazon Comprehend con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Addestra un classificatore Amazon Comprehend personalizzato e classifica i documenti utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Creare un classificatore multietichetta Amazon Comprehend.
- Addestrare il classificatore in base a dati di esempio.
- Eseguire un processo di classificazione su un secondo set di dati.
- Estrarre i dati di output del processo da Amazon S3.

Python

SDK per Python (Boto3)

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Crea una classe wrapper per chiamare le azioni del classificatore di documenti 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

```

Creare una classe a supporto dell'esecuzione dello scenario.

```

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']}"
            f"{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

```

Addestra un classificatore su una serie di GitHub problemi con etichette note, quindi invia un secondo set di GitHub problemi al classificatore in modo che possano essere etichettati.

```

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)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
 - [CreateDocumentClassifier](#)
 - [DeleteDocumentClassifier](#)
 - [DescribeDocumentClassificationJob](#)
 - [DescribeDocumentClassifier](#)
 - [ListDocumentClassificationJobs](#)
 - [ListDocumentClassifiers](#)

- [StartDocumentClassificationJob](#)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di Amazon Comprehend con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in Amazon Comprehend

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#)) lo descrive come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon Comprehend, consulta [AWS Services in Scope by Compliance Program AWS Services in Scope](#) Program.
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon Comprehend. I seguenti argomenti mostrano come configurare Amazon Comprehend per soddisfare i tuoi obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon Comprehend.

Argomenti

- [Protezione dei dati in Amazon Comprehend](#)
- [Identity and Access Management per Amazon Comprehend](#)
- [Registrazione delle chiamate API Amazon Comprehend con AWS CloudTrail](#)
- [Convalida della conformità per Amazon Comprehend](#)
- [Resilienza in Amazon Comprehend](#)
- [Sicurezza dell'infrastruttura in Amazon Comprehend](#)

Protezione dei dati in Amazon Comprehend

Il modello di [responsabilità AWS condivisa modello](#) di si applica alla protezione dei dati in Amazon Comprehend. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [AWS Modello di responsabilità condivisa e GDPR](#) nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon Comprehend o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un

server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Argomenti

- [Crittografia KMS in Amazon Comprehend](#)
- [Prevenzione del confused deputy tra servizi](#)
- [Proteggi i lavori utilizzando un Amazon Virtual Private Cloud](#)
- [Amazon Comprehend e interfaccia gli endpoint VPC \(AWS PrivateLink\)](#)

Crittografia KMS in Amazon Comprehend

Amazon Comprehend collabora con AWS Key Management Service (AWS KMS) per fornire una crittografia avanzata per i tuoi dati. Amazon S3 ti consente già di crittografare i documenti di input quando crei un'analisi del testo, la modellazione di argomenti o un lavoro Amazon Comprehend personalizzato. L'integrazione con AWS KMS consente di crittografare i dati nel volume di storage per i lavori Start* e Create* e crittografa i risultati di output dei lavori Start* utilizzando la tua chiave KMS.

Per il Console di gestione AWS, Amazon Comprehend crittografa i modelli personalizzati con la propria chiave KMS. Inoltre AWS CLI, Amazon Comprehend può crittografare modelli personalizzati utilizzando la propria chiave KMS o una chiave gestita dal cliente (CMK) fornita.

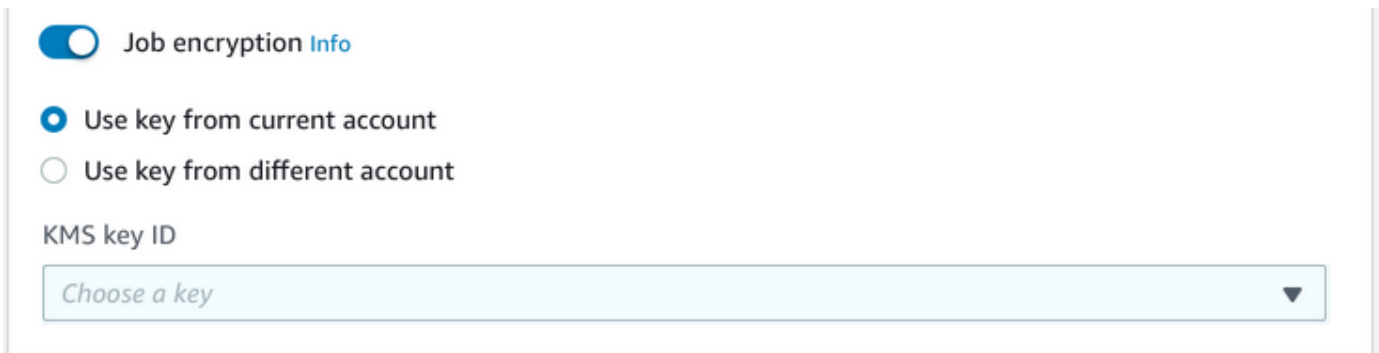
Crittografia KMS utilizzando il Console di gestione AWS

Quando si utilizza la console sono disponibili due opzioni di crittografia:

- Crittografia dei volumi
- crittografia dei risultati di output

Per abilitare la crittografia dei volumi

1. In Job Settings, scegli l'opzione di crittografia del lavoro.



☒ Job encryption [Info](#)

☐ Use key from current account

☒ Use key from different account

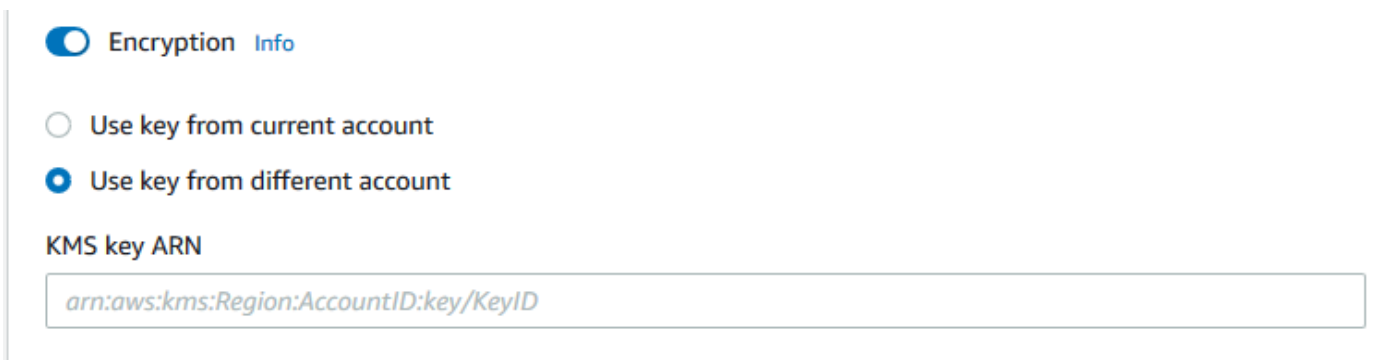
KMS key ID

Choose a key ▼

2. Scegliere se la chiave gestita dal cliente (CMK) KMS è dell'account in uso o di un altro account. Se desideri utilizzare una chiave dell'account corrente, scegli l'alias della chiave da KMS key ID. Se utilizzi una chiave di un altro account, devi inserire l'ARN della chiave.

Per abilitare la crittografia dei risultati di output

1. In Impostazioni di output, scegli l'opzione Crittografia.



☒ Encryption [Info](#)

☐ Use key from current account

☒ Use key from different account

KMS key ARN

arn:aws:kms:Region:AccountID:key/KeyID

2. Scegli se la chiave gestita dal cliente (CMK) proviene dall'account che stai utilizzando attualmente o da un altro account. Se desideri utilizzare una chiave dell'account corrente, scegli l'ID della chiave da KMS key ID. Se utilizzi una chiave di un altro account, devi inserire l'ARN della chiave.

Se in precedenza hai configurato la crittografia utilizzando SSE-KMS sui documenti di input S3, questo può offrirti una sicurezza aggiuntiva. Tuttavia, in tal caso, il ruolo IAM utilizzato deve disporre dell'`kms:Decrypt` autorizzazione per la chiave KMS con cui i documenti di input sono crittografati. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per utilizzare la crittografia KMS](#).

Crittografia KMS con operazioni API

Tutte le operazioni Amazon Comprehend `Start*` e `Create*` API supportano documenti di input crittografati KMS. `Describe*` e le operazioni `List*` API restituiscono l'`KmsKeyId` input

OutputDataConfig se il job originale era stato KmsKeyId fornito come input. Se non è stato fornito come input, non viene restituito.

Questo può essere visto nel seguente esempio di AWS CLI che utilizza l'[StartEntitiesDetectionJob](#) operazione:

```
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/" \  
  --volumekeyid "Volume KMS key ID"
```

Note

Questo esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Crittografia Customer Managed Key (CMK) con operazioni API

Le operazioni API del modello personalizzato di Amazon Comprehend e CreateEntityRecognizer CreateDocumentClassifierCreateEndpoint, supportano la crittografia utilizzando chiavi gestite dal cliente tramite. AWS CLI

È necessaria una policy IAM per consentire a un responsabile di utilizzare o gestire le chiavi gestite dal cliente. Queste chiavi sono specificate nell'Resourceelemento della dichiarazione politica. Come procedura consigliata, limita le chiavi gestite dal cliente solo a quelle che i responsabili devono utilizzare nella dichiarazione sulla politica.

L'esempio AWS CLI seguente crea un riconoscimento di entità personalizzato con crittografia del modello utilizzando l'operazione: [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

Questo esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

Prevenzione del confused deputy tra servizi

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel frattempo AWS, l'impersonificazione tra servizi può portare al confuso problema del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse dell'account.

Ti consigliamo di utilizzare le chiavi di contesto [aws:SourceArn](#) e [aws:SourceAccount](#) global condition nelle politiche delle risorse per limitare le autorizzazioni che Amazon Comprehend fornisce a un altro servizio alla risorsa. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account nella stessa istruzione di policy.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, è necessario utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:servicename::123456789012:*`.

Usare l'account di origine

L'esempio seguente mostra come utilizzare la chiave di contesto della condizione `aws:SourceAccount` globale in Amazon Comprehend.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "comprehend.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
}
```

Policy di fiducia per gli endpoint dei modelli crittografati

È necessario creare una politica di fiducia per creare o aggiornare un endpoint per un modello crittografato. Impostare il valore `aws:SourceAccount` sull'ID dell'account. Se si utilizza la `ArnEquals` condizione, impostare il `aws:SourceArn` valore sull'ARN dell'endpoint.

JSON

```
{
  "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"
      }
    }
  }
]
}

```

Crea un modello personalizzato

È necessario creare una politica di fiducia per creare un modello personalizzato. Impostare il valore `aws:SourceAccount` sull'ID dell'account. Se si utilizza la `ArnEquals` condizione, impostare il `aws:SourceArn` valore sull'ARN della versione del modello personalizzato.

JSON

```

{
  "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"
        }
      }
    }
  ]
}

```

```
}  
}  
}  
}  
}
```

Proteggi i lavori utilizzando un Amazon Virtual Private Cloud

Amazon Comprehend utilizza una serie di misure di sicurezza per garantire la sicurezza dei tuoi dati con i nostri job container in cui vengono archiviati mentre vengono utilizzati da Amazon Comprehend. Tuttavia, i job container accedono a AWS risorse, come i bucket Amazon S3 in cui archiviare dati e modellare artefatti, tramite Internet.

Per controllare l'accesso ai tuoi dati, ti consigliamo di creare un cloud privato virtuale (VPC) e configurarlo in modo che i dati e i contenitori non siano accessibili su Internet. Per ulteriori informazioni sulla creazione e sulla configurazione di un VPC, consulta [Nozioni di base su Amazon VPC](#) nella Guida per l'utente di Amazon VPC. L'utilizzo di un VPC aiuta a proteggere i dati perché è possibile configurare il VPC in modo che non sia connesso a Internet. L'utilizzo di un VPC consente inoltre di monitorare tutto il traffico di rete in entrata e in uscita dai nostri container di lavoro utilizzando i log di flusso VPC. Per ulteriori informazioni, consulta [Log di flusso VPC](#) nella Guida per l'utente di Amazon VPC.

La configurazione del VPC viene specificata quando si crea un lavoro, specificando le sottoreti e i gruppi di sicurezza. Quando specifichi le sottoreti e i gruppi di sicurezza, Amazon Comprehend crea interfacce di rete elastiche ENIs () associate ai tuoi gruppi di sicurezza in una delle sottoreti. ENIs consenti ai nostri container di lavoro di connettersi alle risorse nel tuo VPC. Per informazioni su ENIs, consulta [Elastic Network Interfaces](#) nella Amazon VPC User Guide.

Note

Per i lavori, puoi configurare solo le sottoreti con un VPC di tenancy predefinito in cui l'istanza viene eseguita su hardware condiviso. Per ulteriori informazioni sull'attributo tenancy per VPCs, consulta [Dedicated Instances](#) nella Amazon EC2 User Guide.

Configurare un lavoro per l'accesso ad Amazon VPC

Per specificare sottoreti e gruppi di sicurezza nel tuo VPC, utilizza il parametro di `VpcConfig` richiesta dell'API applicabile o fornisci queste informazioni quando crei un lavoro nella console Amazon Comprehend. Amazon Comprehend utilizza queste informazioni per crearle ENIs e allegarle ai nostri contenitori di lavoro. ENIs Forniamo ai nostri job container una connessione di rete all'interno del tuo VPC che non sia connessa a Internet.

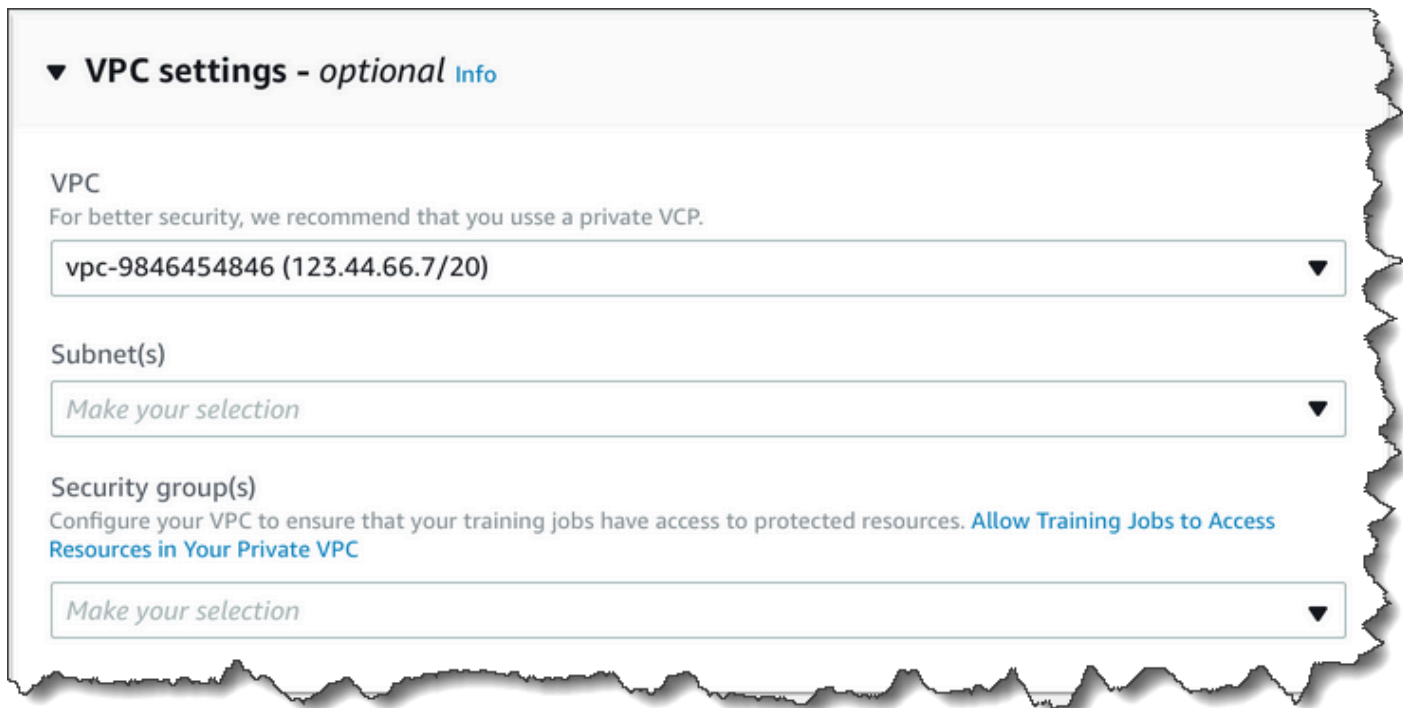
Quanto segue APIs contiene il parametro di `VpcConfig` richiesta:

- Create* APIs: [CreateDocumentClassifier](#), [CreateEntityRecognizer](#)
- Start* APIs: [StartDocumentClassificationJob](#), [StartDominantLanguageDetectionJob](#), [StartEntitiesDetectionJob](#), [StartKeyPhrasesDetectionJob](#), [StartSentimentDetectionJob](#), [StartTargetedSentimentDetectionJob](#), [StartTopicsDetectionJob](#)

Di seguito è riportato un esempio del `VpcConfig` parametro che includi nella chiamata API:

```
"VpcConfig": {  
  "SecurityGroupIds": [  
    " sg-0123456789abcdef0"  
  ],  
  "Subnets": [  
    "subnet-0123456789abcdef0",  
    "subnet-0123456789abcdef1",  
    "subnet-0123456789abcdef2"  
  ]  
}
```

Per configurare un VPC dalla console Amazon Comprehend, scegli i dettagli di configurazione dalla sezione opzionale VPC Settings durante la creazione del lavoro.



▼ **VPC settings - optional** [Info](#)

VPC
For better security, we recommend that you use a private VPC.

vpc-9846454846 (123.44.66.7/20) ▼

Subnet(s)

Make your selection ▼

Security group(s)
Configure your VPC to ensure that your training jobs have access to protected resources. [Allow Training Jobs to Access Resources in Your Private VPC](#)

Make your selection ▼

Configura il tuo VPC per i lavori Amazon Comprehend

Quando configuri il VPC per i tuoi job Amazon Comprehend, utilizza le seguenti linee guida. Per informazioni sulla configurazione di un VPC, consulta [Working with VPCs and Subnet nella Amazon VPC User Guide](#).

Assicurati che le sottoreti abbiano un numero sufficiente di indirizzi IP

Le sottoreti VPC devono avere almeno due indirizzi IP privati per ogni istanza di un job. Per ulteriori informazioni, consulta [VPC and Subnet Sizing nella IPv4 Amazon VPC User Guide](#).

Crea un endpoint VPC Amazon S3

Se configuri il tuo VPC in modo che i job container non abbiano accesso a Internet, non possono connettersi ai bucket Amazon S3 che contengono i tuoi dati a meno che non crei un endpoint VPC che consenta l'accesso. Creando un endpoint VPC, consenti ai job container di accedere ai tuoi dati durante i lavori di formazione e analisi.

Quando crei l'endpoint VPC, configura questi valori:

- Seleziona la categoria di servizio come Servizi AWS
- Specificare il servizio come `com.amazonaws.region.s3`
- Seleziona Gateway come tipo di endpoint VPC

Se stai usando CloudFormation per creare l'endpoint VPC, segui la documentazione. [CloudFormation VPC Endpoint](#) L'esempio seguente mostra la VPC Endpoint configurazione in un CloudFormation modello.

```
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
```

Ti consigliamo anche di creare una policy personalizzata che consenta solo alle richieste provenienti dal tuo VPC di accedere ai tuoi bucket S3. Per ulteriori informazioni, consulta [Endpoint per Amazon S3](#) nella Guida per l'utente di Amazon VPC.

La policy seguente consente l'accesso ai bucket S3. Modifica questa politica per consentire l'accesso solo alle risorse di cui ha bisogno il tuo lavoro.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:DeleteObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "*"
    }
  ]
}
```

Utilizza le impostazioni DNS predefinite per la tabella di routing degli endpoint, in modo che Amazon URLs S3 standard (ad esempio `http://s3-aws-region.amazonaws.com/amzn-s3-demo-bucket`) risolva. Se non utilizzi le impostazioni DNS predefinite, assicurati che quelle utilizzate per specificare le URLS posizioni dei dati nei job vengano risolte configurando le tabelle di routing degli endpoint. Per informazioni sulle tabelle di routing di endpoint VPC, consulta [Routing per endpoint gateway](#) nella Guida per l'utente di Amazon VPC.

La policy predefinita per gli endpoint consente agli utenti di installare pacchetti dai repository Amazon Linux e Amazon Linux 2 nel nostro contenitore dei lavori. Per impedire agli utenti di installare pacchetti da quel repository, crea una policy di endpoint personalizzata che nega esplicitamente l'accesso ai repository di Amazon Linux e Amazon Linux 2. Comprehend di per sé non necessita di tali pacchetti, quindi non ci sarà alcun impatto sulla funzionalità. Di seguito è riportato un esempio di policy che nega l'accesso a questi repository:

```
{
  "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/*"
            ]
        }
    ]
}

```

Autorizzazioni per **DataAccessRole**

Quando si utilizza un VPC per il lavoro di analisi, l'utente `DataAccessRole` utilizzato per le `Start*` operazioni `Create*` and deve disporre anche delle autorizzazioni per il VPC da cui si accede ai documenti di input e al bucket di output.

La seguente politica fornisce l'accesso necessario all'utente `DataAccessRole` per le operazioni `and`. `Create*` `Start*`

JSON

```

{
    "Version": "2012-10-17",

```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "ec2:CreateNetworkInterface",
          "ec2:CreateNetworkInterfacePermission",
          "ec2>DeleteNetworkInterface",
          "ec2>DeleteNetworkInterfacePermission",
          "ec2:DescribeNetworkInterfaces",
          "ec2:DescribeVpcs",
          "ec2:DescribeDhcpOptions",
          "ec2:DescribeSubnets",
          "ec2:DescribeSecurityGroups"
        ],
        "Resource": "*"
      }
    ]
  }
}

```

Configurare il gruppo di sicurezza VPC

Con i lavori distribuiti, è necessario consentire la comunicazione tra i diversi contenitori di lavoro nello stesso lavoro. A tale scopo, configura una regola per il gruppo di sicurezza che consente connessioni in entrata tra i membri dello stesso gruppo di sicurezza. Per informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Guida per l'utente di Amazon VPC.

Connect a risorse esterne al tuo VPC

Se configuri il tuo VPC in modo che non abbia accesso a Internet, i lavori che utilizzano quel VPC non hanno accesso a risorse esterne al tuo VPC. Se i tuoi lavori richiedono l'accesso a risorse esterne al tuo VPC, fornisci l'accesso con una delle seguenti opzioni:

- Se il tuo lavoro richiede l'accesso a un AWS servizio che supporta gli endpoint VPC di interfaccia, crea un endpoint per connetterti a quel servizio. Per un elenco dei servizi che supportano gli endpoint di interfaccia, consulta [Endpoint VPC](#) nella Guida per l'utente di Amazon VPC. Per informazioni sulla creazione di un endpoint VPC di interfaccia, consulta [Interface VPC Endpoints \(\) nella Amazon VPC User AWS PrivateLink Guide](#).
- Se il tuo lavoro richiede l'accesso a un AWS servizio che non supporta gli endpoint VPC di interfaccia o a una risorsa esterna AWS, crea un gateway NAT e configura i tuoi gruppi di sicurezza per consentire le connessioni in uscita. Per informazioni sulla configurazione di un gateway NAT

per il tuo VPC, [consulta Scenario 2: VPC con sottoreti pubbliche e private \(NAT\) nella Amazon VPC User Guide](#).

Amazon Comprehend e interfaccia gli endpoint VPC (AWS PrivateLink)

Puoi stabilire una connessione privata tra il tuo VPC e Amazon Comprehend creando un endpoint VPC di interfaccia. Gli endpoint di interfaccia sono basati su una tecnologia che consente di accedere in modo privato ad Amazon APIs Comprehend senza un gateway Internet, un dispositivo NAT, una connessione VPN o AWS una connessione Direct Connect. [AWS PrivateLink](#) Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare con Amazon Comprehend. APIs Il traffico tra il tuo VPC e Amazon Comprehend non esce dalla rete Amazon.

Ogni endpoint di interfaccia è rappresentato da una o più [interfacce di rete elastiche](#) nelle sottoreti.

Per ulteriori informazioni, consultare [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon VPC.

Considerazioni sugli endpoint VPC Amazon Comprehend

Prima di configurare un endpoint VPC di interfaccia per Amazon Comprehend, assicurati di esaminare le proprietà [e le limitazioni degli endpoint dell'interfaccia nella Amazon VPC User Guide](#).

Gli endpoint Amazon Comprehend non sono disponibili in tutte le zone di disponibilità di una regione. Quando crei l'endpoint, usa il seguente comando per elencare le zone di disponibilità.

```
aws ec2 describe-vpc-endpoint-services \
  --service-names com.amazonaws.us-west-2.comprehend
```

Amazon Comprehend supporta l'esecuzione di chiamate a tutte le sue azioni API dal tuo VPC.

Creazione di un endpoint VPC di interfaccia per Amazon Comprehend

Puoi creare un endpoint VPC per il servizio Amazon Comprehend utilizzando la console Amazon VPC o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consultare [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC.

Crea un endpoint VPC per Amazon Comprehend utilizzando il seguente nome di servizio:

- com.amazonaws. *region*. comprehendere

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API ad Amazon Comprehend utilizzando il nome DNS predefinito per la regione, ad esempio. *comprehend.us-east-1.amazonaws.com*

Per ulteriori informazioni, consultare [Accesso a un servizio tramite un endpoint di interfaccia](#) in Guida per l'utente di Amazon VPC.

Creazione di una policy sugli endpoint VPC per Amazon Comprehend

Puoi allegare una policy per gli endpoint al tuo endpoint VPC che controlla l'accesso ad Amazon Comprehend. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) in Guida per l'utente di Amazon VPC.

Esempio: policy sugli endpoint VPC per le azioni di Amazon Comprehend

Di seguito è riportato un esempio di policy sugli endpoint per Amazon Comprehend. Se collegata a un endpoint, questa policy garantisce l'accesso all'azione Amazon DetectEntities Comprehend per tutti i principali su tutte le risorse.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "comprehend:DetectEntities"
      ],
      "Resource": "*"
    }
  ]
}
```

Identity and Access Management per Amazon Comprehend

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuare l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse Amazon Comprehend. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Come funziona Amazon Comprehend con IAM](#)
- [Esempi di policy basate sull'identità per Amazon Comprehend](#)
- [AWS politiche gestite per Amazon Comprehend](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Comprehend](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Comprehend](#))
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta [Come funziona Amazon Comprehend con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta [Esempi di policy basate sull'identità per Amazon Comprehend](#))

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/

Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Come funziona Amazon Comprehend con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon Comprehend, scopri quali funzionalità IAM sono disponibili per l'uso con Amazon Comprehend.

Funzionalità IAM che puoi usare con Amazon Comprehend

Funzionalità IAM	Supporto per Amazon Comprehend
Policy basate sull'identità	Sì
Policy basate su risorse	Sì
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come Amazon Comprehend e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, [AWS consulta i servizi che funzionano con](#) IAM nella IAM User Guide.

Politiche basate sull'identità per Amazon Comprehend

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Esempi di policy basate sull'identità per Amazon Comprehend

Per visualizzare esempi di politiche basate sull'identità di Amazon Comprehend, consulta. [Esempi di policy basate sull'identità per Amazon Comprehend](#)

Politiche basate sulle risorse all'interno di Amazon Comprehend

Supporta le policy basate sulle risorse: sì

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Il servizio Amazon Comprehend supporta solo un tipo di politica basata sulle risorse (una politica del modello personalizzata), collegata a un modello personalizzato. Questa politica definisce altri account che possono utilizzare il modello personalizzato.

Per informazioni su come allegare una politica basata sulle risorse a un modello personalizzato, consulta. [Politiche basate sulle risorse per modelli personalizzati](#)

Azioni politiche per Amazon Comprehend

Supporta le operazioni di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di azioni Amazon Comprehend, consulta [Actions defined by Amazon Comprehend](#) nel Service Authorization Reference.

Le azioni politiche in Amazon Comprehend utilizzano il seguente prefisso prima dell'azione:

```
comprehend
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
    "comprehend:DetectSentiment",  
    "comprehend:ClassifyDocument"  
]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "comprehend:Describe*"
```

Non utilizzare caratteri jolly per specificare tutte le azioni di un servizio. Utilizza la migliore pratica di concedere il privilegio minimo quando specifichi le autorizzazioni in una politica.

Per visualizzare esempi di politiche basate sull'identità di Amazon Comprehend, consulta [Esempi di policy basate sull'identità per Amazon Comprehend](#)

Risorse relative alle policy per Amazon Comprehend

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di risorse Amazon Comprehend e relativi ARNs, consulta [Resources defined by Amazon Comprehend](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta [Azioni definite da Amazon Comprehend](#).

Chiavi delle condizioni delle politiche per Amazon Comprehend

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco di chiavi di condizione di Amazon Comprehend, consulta [Condition keys for Amazon Comprehend](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Azioni definite da Amazon Comprehend](#).

Per visualizzare esempi di politiche basate sull'identità di Amazon Comprehend, consulta [Esempi di policy basate sull'identità per Amazon Comprehend](#)

ACLs in Amazon Comprehend

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Amazon Comprehend

Supporta ABAC (tag nelle policy): parzialmente

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Per ulteriori informazioni sull'etichettatura delle risorse Amazon Comprehend, consulta. [Tagging delle risorse](#)

Utilizzo di credenziali temporanee con Amazon Comprehend

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono un accesso a breve termine alle AWS risorse e vengono create automaticamente quando utilizzi la federazione o cambi ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

Sessioni di accesso diretto per Amazon Comprehend

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale chiamante e Servizio AWS, in combinazione con la richiesta, Servizio AWS per effettuare richieste ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli di servizio per Amazon Comprehend

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di Amazon Comprehend. Modifica i ruoli di servizio solo quando Amazon Comprehend fornisce indicazioni in tal senso.

Per utilizzare le operazioni asincrone di Amazon Comprehend, devi concedere ad Amazon Comprehend l'accesso al bucket Amazon S3 che contiene la tua raccolta di documenti. A tale scopo, puoi creare un ruolo di accesso ai dati nel tuo account con una politica di fiducia per considerare affidabile il responsabile del servizio Amazon Comprehend.

Per un esempio di policy, vedi [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#)

Ruoli collegati ai servizi per Amazon Comprehend

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per Amazon Comprehend

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse Amazon Comprehend. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da Amazon Comprehend, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Azioni, risorse e chiavi di condizione per Amazon Comprehend](#) nel Service Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console Amazon Comprehend](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Autorizzazioni necessarie per eseguire azioni di analisi dei documenti](#)
- [Autorizzazioni necessarie per utilizzare la crittografia KMS](#)
- [AWS politiche gestite \(predefinite\) per Amazon Comprehend](#)
- [Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone](#)
- [Autorizzazioni per consentire tutte le azioni di Amazon Comprehend](#)
- [Autorizzazioni per consentire azioni di modellazione degli argomenti](#)
- [Autorizzazioni necessarie per un processo di analisi asincrono personalizzato](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Comprehend nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le politiche AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti

specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.

- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console Amazon Comprehend

Per accedere alla console Amazon Comprehend, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon Comprehend presenti nel tuo Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per le autorizzazioni minime della console Amazon Comprehend, puoi allegare la policy *ComprehendReadOnly* AWS gestita alle entità. Per maggiori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente di IAM.

Per utilizzare la console Amazon Comprehend, sono inoltre necessarie le autorizzazioni per le azioni mostrate nella seguente politica:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:ListRoles",
        "iam:GetRole",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

La console Amazon Comprehend necessita di queste autorizzazioni aggiuntive per i seguenti motivi:

- iam autorizzazioni per elencare i ruoli IAM disponibili per il tuo account.
- s3 autorizzazioni per accedere ai bucket e agli oggetti Amazon S3 che contengono i dati per la modellazione degli argomenti.

Quando crei un processo batch asincrono o un lavoro di modellazione di argomenti utilizzando la console, hai la possibilità di fare in modo che la console crei un ruolo IAM per il tuo lavoro. Per creare

un ruolo IAM, agli utenti devono essere concesse le seguenti autorizzazioni aggiuntive per creare ruoli e policy IAM e per associare policy ai ruoli:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/*Comprehend*"
    }
  ]
}
```

La console Amazon Comprehend necessita di queste autorizzazioni aggiuntive per i seguenti motivi:

- `iam` autorizzazioni per creare ruoli e politiche e per allegare ruoli e politiche.
L'`iam:PassRole` azione consente alla console di passare il ruolo ad Amazon Comprehend.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica

include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"
    }
  ]
}
```

Autorizzazioni necessarie per eseguire azioni di analisi dei documenti

La seguente politica di esempio concede le autorizzazioni per utilizzare le azioni di analisi dei documenti di Amazon Comprehend:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowDetectActions",
    "Effect": "Allow",
    "Action": [
      "comprehend:DetectEntities",
      "comprehend:DetectKeyPhrases",
      "comprehend:DetectDominantLanguage",
      "comprehend:DetectSentiment",
      "comprehend:DetectTargetedSentiment",
      "comprehend:DetectSyntax",
      "textract:DetectDocumentText",
      "textract:AnalyzeDocument"
    ],
    "Resource": "*"
  }]
}
```

La policy contiene una dichiarazione che concede il permesso di utilizzare le azioni `DetectEntities`, `DetectKeyPhrases`, `DetectDominantLanguage`, `DetectTargetedSentiment`, `DetectSentiment` e `DetectSyntax`. La dichiarazione politica concede inoltre le autorizzazioni per l'uso di due metodi API Amazon Textract. Amazon Comprehend utilizza questi metodi per estrarre testo da file di immagini e documenti PDF scansionati. Puoi rimuovere queste autorizzazioni per gli utenti che non eseguono mai inferenze personalizzate per questi tipi di file di input.

Un utente con questa politica non sarebbe in grado di eseguire azioni batch o azioni asincrone nel proprio account.

La policy non specifica l'elemento `Principal` poiché in una policy basata su identità non si specifica il principale che ottiene l'autorizzazione. Quando allegghi una policy a un utente, l'utente è il principale implicito. Quando colleghi una policy di autorizzazioni a un ruolo IAM, il principale identificato nella policy di attendibilità del ruolo ottiene le autorizzazioni.

Per una tabella che mostra tutte le azioni dell'API Amazon Comprehend e le risorse a cui si applicano, consulta [Azioni, risorse e chiavi di condizione per Amazon Comprehend](#) nel Service Authorization Reference.

Autorizzazioni necessarie per utilizzare la crittografia KMS

Per utilizzare appieno Amazon Key Management Service (KMS) per la crittografia di dati e processi in un processo asincrono, devi concedere le autorizzazioni per le azioni mostrate nella seguente politica:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:CreateGrant"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDatakey"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": [
            "s3.us-east-1.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

Quando crei un lavoro asincrono con Amazon Comprehend, usi i dati di input archiviati su Amazon S3. Con S3, hai la possibilità di crittografare i dati archiviati, che vengono crittografati da S3 e non da Amazon Comprehend. Possiamo decrittografare e leggere i dati di input crittografati se fornisci l'`kms:Decrypt` autorizzazione per la chiave con cui i dati di input originali sono stati crittografati per il ruolo di accesso ai dati utilizzato dal job Amazon Comprehend.

Hai anche la possibilità di utilizzare le chiavi gestite dal cliente (CMK) KMS per crittografare i risultati di output su S3, nonché il volume di storage utilizzato durante l'elaborazione del lavoro. A tale scopo, è possibile utilizzare la stessa chiave KMS per entrambi i tipi di crittografia, ma ciò non è necessario. Durante la creazione del lavoro sono disponibili campi separati per specificare le chiavi per la crittografia dell'output e la crittografia del volume ed è anche possibile utilizzare una chiave KMS di un account diverso.

Quando si utilizza la crittografia KMS, è richiesta `kms:CreateGrant` l'autorizzazione per la crittografia dei volumi e `kms:GenerateDataKey` l'autorizzazione è necessaria per la crittografia dei dati di output. Per leggere input crittografati (come quando i dati di input sono già crittografati da Amazon S3), è `kms:Decrypt` richiesta l'autorizzazione. Il ruolo IAM deve fornire queste autorizzazioni secondo necessità. Tuttavia, se la chiave proviene da un account diverso da quello attualmente utilizzato, la politica delle chiavi KMS per quella chiave `kms` deve fornire queste autorizzazioni anche al ruolo di accesso ai dati per il lavoro.

AWS politiche gestite (predefinite) per Amazon Comprehend

AWS affronta molti casi d'uso comuni fornendo politiche IAM autonome create e amministrate da AWS. Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni in modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consultare [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche di Amazon Comprehend:

- **ComprehendFullAccess**— Garantisce l'accesso completo alle risorse di Amazon Comprehend, inclusa l'esecuzione di lavori di modellazione tematica. Include il permesso di elencare e ottenere ruoli IAM.
- **ComprehendReadOnly**— Concede l'autorizzazione a eseguire tutte le azioni di Amazon Comprehend `StartDominantLanguageDetectionJob` tranne `StartEntitiesDetectionJob`, `StartKeyPhrasesDetectionJob`, `StartSentimentDetectionJob` `StartTargetedSentimentDetectionJob`, e `StartTopicsDetectionJob`.

È necessario applicare la seguente politica aggiuntiva a tutti gli utenti che utilizzeranno Amazon Comprehend:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/*Comprehend*"
    }
  ]
}
```

Puoi rivedere le politiche di autorizzazione gestite accedendo alla console IAM e cercando lì politiche specifiche.

Queste politiche funzionano quando si utilizza AWS SDKs o la AWS CLI.

Puoi anche creare policy IAM personalizzate per consentire le autorizzazioni per azioni e risorse di Amazon Comprehend. Puoi allegare queste politiche personalizzate agli utenti, ai gruppi o ai ruoli che richiedono tali autorizzazioni.

Autorizzazioni basate sui ruoli necessarie per le operazioni asincrone

Per utilizzare le operazioni asincrone di Amazon Comprehend, devi concedere ad Amazon Comprehend l'accesso al bucket Amazon S3 che contiene la tua raccolta di documenti. A tale scopo, puoi creare un ruolo di accesso ai dati nel tuo account con una politica di fiducia per considerare affidabile il responsabile del servizio Amazon Comprehend. Per ulteriori informazioni sulla creazione di un ruolo, vedere [Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio](#) nella Guida per l'utente di AWS Identity and Access Management.

Di seguito viene illustrato un esempio di politica di fiducia per il ruolo creato. Per facilitare la [prevenzione della confusione tra i deputati](#), limitate l'ambito dell'autorizzazione utilizzando una o più

chiavi di contesto relative alle condizioni globali. Impostare il valore `aws:SourceAccount` sull'ID dell'account. Se si utilizza la `ArnEquals` condizione, impostare il `aws:SourceArn` valore sull'ARN del lavoro. Usa un carattere jolly per il numero del lavoro nell'ARN, poiché Amazon Comprehend genera questo numero come parte della creazione di posti di lavoro.

JSON

```
{
  "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/*"
        }
      }
    }
  ]
}
```

Dopo aver creato il ruolo, crea una politica di accesso per quel ruolo. Ciò dovrebbe concedere Amazon S3 `GetObject` e `ListBucket` le autorizzazioni al bucket Amazon S3 che contiene i dati di input e l'autorizzazione `PutObject` Amazon S3 al bucket di dati di output Amazon S3.

Autorizzazioni per consentire tutte le azioni di Amazon Comprehend

Dopo la registrazione AWS, crei un utente amministratore per gestire il tuo account, inclusa la creazione di utenti e la gestione delle relative autorizzazioni.

Puoi scegliere di creare un utente con le autorizzazioni per tutte le azioni di Amazon Comprehend (considera questo utente un amministratore specifico del servizio) per lavorare con Amazon Comprehend. È possibile collegare la policy di autorizzazione seguente a tale utente.

JSON

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

Queste autorizzazioni possono essere modificate per quanto riguarda la crittografia nei seguenti modi:

- Per consentire ad Amazon Comprehend di analizzare i documenti archiviati in un bucket S3 crittografato, il ruolo IAM deve disporre dell'autorizzazione. `kms:Decrypt`
- Per consentire ad Amazon Comprehend di crittografare i documenti archiviati su un volume di storage collegato all'istanza di calcolo che elabora il processo di analisi, il ruolo IAM deve disporre dell'autorizzazione. `kms:CreateGrant`
- Per consentire ad Amazon Comprehend di crittografare i risultati di output nel proprio bucket S3, il ruolo IAM deve disporre dell'autorizzazione. `kms:GenerateDataKey`

Autorizzazioni per consentire azioni di modellazione degli argomenti

La seguente politica di autorizzazione concede le autorizzazioni agli utenti per eseguire le operazioni di modellazione tematica di Amazon Comprehend.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTopicModelingActions",
      "Effect": "Allow",
      "Action": [
        "comprehend:DescribeTopicsDetectionJob",
        "comprehend:ListTopicsDetectionJobs",
        "comprehend:StartTopicsDetectionJob"
      ],
      "Resource": "*"
    }
  ]
}
```

Autorizzazioni necessarie per un processo di analisi asincrono personalizzato

Important

Se disponi di una policy IAM che limita l'accesso al modello, non sarai in grado di completare un lavoro di inferenza con un modello personalizzato. La tua policy IAM dovrebbe essere aggiornata in modo da avere una risorsa wildcard per un processo di analisi asincrona personalizzato.

Se utilizzi [StartDocumentClassificationJob](#) and [StartEntitiesDetectionJob](#) APIs, devi aggiornare la tua policy IAM a meno che non utilizzi attualmente i caratteri jolly come risorse. Se [StartEntitiesDetectionJob](#) utilizzi un modello già addestrato, ciò non ha alcun impatto su di te e non devi apportare alcuna modifica.

La seguente politica di esempio contiene un riferimento obsoleto.

```
{
  "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"
}
```

Questa è la politica aggiornata che è necessario utilizzare per eseguire correttamente and. `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 politiche gestite per Amazon Comprehend

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite che scriverle da soli. La [creazione di policy gestite dai clienti IAM](#) che forniscono al team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste policy coprono i casi d'uso comuni e sono disponibili nell'account Account AWS. Per ulteriori informazioni sulle policy AWS gestite, consulta le [policy AWS gestite](#) nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consultare la sezione [Policy gestite da AWS per funzioni di processi](#) nella Guida per l'utente di IAM.

AWS politica gestita: ComprehendFullAccess

Questa politica garantisce l'accesso completo alle risorse di Amazon Comprehend, inclusa l'esecuzione di lavori di modellazione degli argomenti. Questa policy concede inoltre autorizzazioni per l'elenco e l'ottenimento di autorizzazioni per i bucket Amazon S3 e i ruoli IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "comprehend:*",
        "iam:GetRole",
        "iam:ListRoles",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS politica gestita: ComprehendReadOnly

Questa politica concede autorizzazioni di sola lettura per eseguire tutte le azioni di Amazon Comprehend tranne le seguenti:

- StartDominantLanguageDetectionJob
- StartEntitiesDetectionJob
- StartKeyPhrasesDetectionJob
- StartSentimentDetectionJob
- StartTargetedSentimentDetectionJob
- StartTopicsDetectionJob

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Action": [
    "comprehend:BatchDetectDominantLanguage",
    "comprehend:BatchDetectEntities",
    "comprehend:BatchDetectKeyPhrases",
    "comprehend:BatchDetectSentiment",
    "comprehend:BatchDetectSyntax",
    "comprehend:ClassifyDocument",
    "comprehend:ContainsPiiEntities",
    "comprehend:DescribeDocumentClassificationJob",
    "comprehend:DescribeDocumentClassifier",
    "comprehend:DescribeDominantLanguageDetectionJob",
    "comprehend:DescribeEndpoint",
    "comprehend:DescribeEntitiesDetectionJob",
    "comprehend:DescribeEntityRecognizer",
    "comprehend:DescribeKeyPhrasesDetectionJob",
    "comprehend:DescribePiiEntitiesDetectionJob",
    "comprehend:DescribeResourcePolicy",
    "comprehend:DescribeSentimentDetectionJob",
    "comprehend:DescribeTargetedSentimentDetectionJob",
    "comprehend:DescribeTopicsDetectionJob",
    "comprehend:DetectDominantLanguage",
    "comprehend:DetectEntities",
    "comprehend:DetectKeyPhrases",
    "comprehend:DetectPiiEntities",
    "comprehend:DetectSentiment",
    "comprehend:DetectSyntax",
    "comprehend:ListDocumentClassificationJobs",
    "comprehend:ListDocumentClassifiers",
    "comprehend:ListDocumentClassifierSummaries",
    "comprehend:ListDominantLanguageDetectionJobs",
    "comprehend:ListEndpoints",
    "comprehend:ListEntitiesDetectionJobs",
    "comprehend:ListEntityRecognizers",
    "comprehend:ListEntityRecognizerSummaries",
    "comprehend:ListKeyPhrasesDetectionJobs",
    "comprehend:ListPiiEntitiesDetectionJobs",
    "comprehend:ListSentimentDetectionJobs",
    "comprehend:ListTargetedSentimentDetectionJobs",
    "comprehend:ListTagsForResource",
    "comprehend:ListTopicsDetectionJobs"
  ],
  "Effect": "Allow",
  "Resource": "*"
}

```

```
}
    ]
}
```

Amazon Comprehend: aggiornamenti alle AWS politiche gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon Comprehend da quando questo servizio ha iniziato a tracciare queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti di Amazon [Comprehend](#).

Modifica	Descrizione	Data
ComprehendReadOnly : aggiornamento a una policy esistente	Amazon Comprehend ora consente le comprehension:ListTargetedSentimentDetectionJobs azioni comprehension:DescribeTargetedSentimentDetectionJob e le azioni contenute nella policy ComprehendReadOnly	30 marzo 2022
ComprehendReadOnly : aggiornamento di una policy esistente	Amazon Comprehend ora consente l'comprehend:DescribeResourcePolicy azione nella politica ComprehendReadOnly	2 febbraio 2022
ComprehendReadOnly : aggiornamento di una policy esistente	Amazon Comprehend ora consente le ListEntityRecognizerSummaries azioni ListDocumentClassifierSummaries e le azioni contenute	21 settembre 2021

Modifica	Descrizione	Data
	nella policy ComprehendReadOnly	
ComprehendReadOnly : aggiornamento di una policy esistente	Amazon Comprehend ora consente l'ContainsP IIEntities azione nella politica ComprehendReadOnly	26 marzo 2021
Amazon Comprehend ha iniziato a tracciare le modifiche	Amazon Comprehend ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	1° marzo 2021

Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Comprehend

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon Comprehend e IAM.

Argomenti

- [Non sono autorizzato a eseguire un'azione in Amazon Comprehend](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Desidero consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon Comprehend](#)

Non sono autorizzato a eseguire un'azione in Amazon Comprehend

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente mateojackson IAM prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *my-example-widget* fittizia ma non dispone di autorizzazioni comprehend: *GetWidget* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
comprehend:GetWidget on resource: my-example-widget
```

In questo caso, la policy deve essere aggiornata in modo che Mateo possa accedere alla risorsa *my-example-widget* mediante l'operazione comprehend: *GetWidget*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'iam:PassRoleazione, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo ad Amazon Comprehend.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato marymajor tenta di utilizzare la console per eseguire un'azione in Amazon Comprehend. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione iam:PassRole.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon Comprehend

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo.

Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon Comprehend supporta queste funzionalità, consulta [Come funziona Amazon Comprehend con IAM](#)
- Per sapere come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Registrazione delle chiamate API Amazon Comprehend con AWS CloudTrail

Amazon Comprehend è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Amazon Comprehend. CloudTrail acquisisce le chiamate API per Amazon Comprehend come eventi. Le chiamate acquisite includono chiamate dalla console Amazon Comprehend e chiamate in codice alle operazioni dell'API Amazon Comprehend. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon Comprehend. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon Comprehend, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la [Guida per l'AWS CloudTrail utente](#).

Amazon Comprehend: informazioni in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività di evento supportata in Amazon Comprehend, tale attività viene registrata in

un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti in. Account AWS Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi del tuo sito Account AWS, compresi gli eventi per Amazon Comprehend, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il percorso registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, è possibile configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Amazon Comprehend supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- [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](#)

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Esempio: voci dei file di log di Amazon Comprehend

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni

sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'ClassifyDocumentazione.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AROAI CFHPEXAMPLE",
    "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"  
}
```

Convalida della conformità per Amazon Comprehend

I revisori di terze parti valutano la sicurezza e la conformità di Amazon Comprehend nell'ambito di AWS diversi programmi di conformità. Questi includono PCI, FedRAMP, HIPAA e altri. Puoi scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#).

La tua responsabilità di conformità quando usi Amazon Comprehend è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono i passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- [Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper](#) descrive in che modo le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- [AWS Risorse per la conformità](#): questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operi.
- [AWS Config](#)— Questo AWS servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub CSPM](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente AWS di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta la sezione [AWS Servizi rientranti nell'ambito del programma di conformità](#). Per informazioni generali, consulta [Programmi di conformità di AWS](#).

Resilienza in Amazon Comprehend

L'infrastruttura AWS globale è costruita attorno a Regione AWS e alle zone di disponibilità. Regione AWS Le fornisce più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

[Per ulteriori informazioni su Regione AWS e sulle zone di disponibilità, consulta Global Infrastructure.AWS](#)

Sicurezza dell'infrastruttura in Amazon Comprehend

In quanto servizio gestito, Amazon Comprehend è protetto dalla sicurezza della rete AWS globale. Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizza chiamate API AWS pubblicate per accedere ad Amazon Comprehend attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Linee guida e quote

Se non diversamente specificato, le quote di Amazon Comprehend si intendono per regione. Puoi richiedere un aumento delle quote regolabili se necessario per le tue applicazioni. Per informazioni sulle quote e per richiedere un aumento delle quote, vedere [AWS Service Quotas](#).

Argomenti

- [Regioni supportate](#)
- [Quote per i modelli integrati](#)
- [Quote per modelli personalizzati](#)
- [Quote per i volani](#)

Regioni supportate

Amazon Comprehend è disponibile nelle seguenti regioni: AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- AWS GovCloud (Stati Uniti occidentali)

Per impostazione predefinita, Amazon Comprehend fornisce tutte le operazioni API in ciascuna delle regioni supportate. Per le eccezioni, consulta [Elaborazione dei documenti](#)

Per informazioni sugli endpoint delle API, consulta [Amazon Comprehend Regions and Endpoints](#) nel Amazon Web Services General Reference.

Per rivedere le quote correnti in una regione o per richiedere aumenti delle quote per quote regolabili, apri la console [Service Quotas](#).

Quote per i modelli integrati

Amazon Comprehend fornisce modelli integrati per analizzare documenti di testo UTF-8. Amazon Comprehend fornisce operazioni sincrone e asincrone che utilizzano i modelli integrati.

Argomenti

- [Analisi \(sincrona\) in tempo reale](#)
- [Analisi asincrona](#)

Analisi (sincrona) in tempo reale

Questa sezione descrive le quote relative all'analisi in tempo reale utilizzando i modelli integrati.

Argomenti

- [Operazioni su un singolo documento](#)
- [Operazioni su più documenti](#)
- [Limitazione delle richieste per richieste in tempo reale \(sincrone\)](#)

Operazioni su un singolo documento

L'API Amazon Comprehend fornisce operazioni che richiedono un singolo documento come input. Le seguenti quote si applicano a queste operazioni.

Quote generali per le operazioni relative a singoli documenti

Le seguenti quote si applicano all'analisi in tempo reale per il rilevamento di entità, frasi chiave o lingua dominante. Per il rilevamento delle entità, queste quote si applicano al rilevamento con i modelli integrati. Per il rilevamento personalizzato delle entità, consulta le quote in. [Riconoscimento personalizzato delle entità](#)

Description	Quota/Linea guida
Dimensione massima del documento	100 KB

Quote specifiche per le operazioni relative a singoli documenti

Le seguenti quote si applicano all'analisi in tempo reale per rilevare il sentimento, il sentimento mirato e la sintassi.

Description	Quota/Linea guida
Dimensione massima del documento	5 KB

Operazioni su più documenti

L'API Amazon Comprehend fornisce operazioni in batch che elaborano più documenti con un'unica richiesta API. Le seguenti quote si applicano alle operazioni batch.

Description	Quota/linea guida
Dimensione massima del documento	5 KB
Numero massimo di documenti per richiesta	25

Per ulteriori informazioni sull'utilizzo delle operazioni relative ai documenti batch, vedere [Elaborazione sincrona di più documenti](#).

Limitazione delle richieste per richieste in tempo reale (sincrone)

Amazon Comprehend applica la limitazione dinamica alle richieste sincrone. Se la larghezza di banda per l'elaborazione del sistema è disponibile, Amazon Comprehend aumenta gradualmente il numero di richieste elaborate. Per controllare l'utilizzo da parte dell'applicazione delle operazioni API sincrone, ti consigliamo di attivare gli avvisi di fatturazione o di implementare la limitazione della velocità nell'applicazione.

Analisi asincrona

Questa sezione descrive le quote relative all'analisi asincrona utilizzando i modelli integrati.

Le operazioni API asincrone supportano ciascuna un massimo di 10 lavori attivi. Per visualizzare le quote per ogni operazione API, consulta la tabella Service Quotas negli [endpoint Amazon Comprehend e le quote nell'Amazon](#) Web Services General Reference.

Per le quote regolabili, puoi richiedere un aumento della quota utilizzando la console [Service Quotas](#).

Argomenti

- [Quote generali per operazioni asincrone](#)
- [Quote specifiche dell'operazione per lavori asincroni](#)
- [Limitazione delle richieste per richieste asincrone](#)

Quote generali per operazioni asincrone

È possibile eseguire lavori di analisi asincroni utilizzando la console o una qualsiasi delle operazioni API. **Start*** Per informazioni su quando utilizzare le operazioni asincrone, vedere. [Elaborazione asincrona in batch](#) Le seguenti quote si applicano alla maggior parte delle **Start*** operazioni API per i modelli integrati. Per le eccezioni, vedere. [Quote specifiche dell'operazione per lavori asincroni](#)

Description	Quota/Linea guida
Dimensione massima di ogni documento nei lavori che rilevano entità, frasi chiave, PII e lingue	1 MB
Dimensione totale massima di tutti i file in una richiesta	5 GB
Dimensione totale minima di tutti i file in una richiesta	500 byte
Numero massimo di file, un documento per file	1.000.000
Numero massimo totale di righe, un documento per riga	1.000.000

Quote specifiche dell'operazione per lavori asincroni

Questa sezione descrive le quote per operazioni asincrone specifiche. Se una quota non è specificata nelle tabelle seguenti, si applica il valore generale della quota.

Argomenti

- [Sentiment](#)
- [Sentimento mirato](#)
- [Eventi](#)
- [Modellazione degli argomenti](#)

Sentiment

I lavori di sentiment asincroni, creati con l'[StartSentimentDetectionJob](#) operazione, hanno le seguenti quote.

Description	Quota/Linea guida
Dimensione massima di ogni documento di input	5 KB

Sentimento mirato

I job asincroni mirati basati sul sentiment, creati con l'[StartTargetedSentimentDetectionJob](#) operazione, hanno le seguenti quote.

Description	Quota/Linea guida
Formati di documenti supportati	UTF-8
Dimensione massima di ogni documento in un lavoro	10 KB
Dimensione massima di tutti i documenti di un lavoro	300 MB
Numero massimo di file, un documento per file	30.000
Numero massimo totale di righe, un documento per riga (per tutti i file di una richiesta)	30.000

Eventi

I job asincroni di rilevamento degli eventi, creati con l'[StartEventsDetectionJob](#) operazione, hanno le seguenti quote.

Description	Quote
Codifica caratteri	UTF-8
Dimensione totale di tutti i file in un lavoro	50 MB
Dimensione massima di ogni documento in un lavoro	10 KB
Numero massimo di file, un documento per file	5.000
Numero massimo totale di righe, un documento per riga (per tutti i file richiesti)	5.000

Modellazione degli argomenti

I lavori di modellazione asincrona degli argomenti, creati con l'[StartTopicsDetectionJob](#) operazione, hanno le seguenti quote.

Description	Quota/Linea guida
Codifica caratteri	UTF-8
Numero massimo di argomenti da restituire	100
Dimensione massima per un file, un documento per file	100 MB

Per ulteriori informazioni, consulta [Modellazione degli argomenti](#)

Limitazione delle richieste per richieste asincrone

Ogni operazione API asincrona supporta un numero massimo di richieste al secondo (per regione, per account) e anche un massimo di 10 lavori attivi. Per visualizzare le quote per ogni operazione API, consulta la tabella Service Quotas negli [endpoint Amazon Comprehend e le quote nell'Amazon Web Services General Reference](#).

Per le quote regolabili, puoi richiedere un aumento della quota utilizzando la console [Service Quotas](#).

Quote per modelli personalizzati

Puoi usare Amazon Comprehend per creare modelli personalizzati per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Questa sezione fornisce le linee guida e le quote relative alla formazione e all'utilizzo di modelli personalizzati. Per ulteriori informazioni sui modelli personalizzati, vedere [Amazon Comprehend personalizzato](#).

Argomenti

- [Quote generali](#)
- [Quote per gli endpoint](#)
- [Classificazione dei documenti](#)
- [Riconoscimento personalizzato delle entità](#)

Quote generali

Amazon Comprehend stabilisce quote di dimensione generali per ogni tipo di documento di input che puoi analizzare con modelli personalizzati. Per le quote di analisi in tempo reale, consulta. [Dimensioni massime dei documenti per l'analisi in tempo reale](#) Per le quote di analisi asincrone, vedere. [Ingressi per analisi asincrone personalizzate](#)

Ogni operazione API asincrona supporta un numero massimo di richieste al secondo (per regione, per account) e anche un massimo di 10 lavori attivi. Per visualizzare le quote per ogni operazione API, consulta la tabella Service Quotas negli [endpoint Amazon Comprehend e le quote nell'Amazon Web Services General Reference](#).

Per le quote regolabili, puoi richiedere un aumento della quota utilizzando la console [Service Quotas](#).

Quote per gli endpoint

Crei un endpoint per eseguire analisi in tempo reale con un modello personalizzato. Per informazioni sugli endpoint, consulta. [Gestione degli endpoint Amazon Comprehend](#)

Le seguenti quote si applicano agli endpoint. Per informazioni su come richiedere un aumento della quota, vedere [AWS Service Quotas](#).

Description	Quota/Linea guida
Numero massimo di endpoint attivi per regione per ogni account	20
Numero massimo di unità di inferenza per regione per ogni account	200
Numero massimo di unità di inferenza per endpoint per regione	50
Velocità effettiva massima per unità di inferenza (caratteri)	100 al secondo
Velocità effettiva massima per unità di inferenza (documenti)	2/secondo

Classificazione dei documenti

Questa sezione descrive le linee guida e le quote per le seguenti operazioni di classificazione dei documenti:

- Classificate i lavori di formazione che iniziate con l'[CreateDocumentClassifier](#) operazione.
- Processi asincroni di classificazione dei documenti avviati con l'operazione. [StartDocumentClassificationJob](#)
- Richieste di classificazione sincrona dei documenti che utilizzano l'operazione. [ClassifyDocument](#)

Quote generali per la classificazione dei documenti

La tabella seguente descrive le quote generali relative alla formazione di classificatori personalizzati.

Description	Quota/linea guida
Lunghezza massima del nome della classe	5.000 caratteri
Numero di classi (modalità multiclasse)	2—1.000
Numero di classi (modalità multietichetta)	2—100
Formato delle annotazioni	
Numero minimo di annotazioni per classe (modalità multiclasse)	10

Description	Quota/linea guida
Numero minimo di annotazioni per classe (modalità multi-etichetta)	10
Numero minimo di annotazioni (modalità multietichetta)	50
Formato di file CSV	
Numero minimo di documenti di formazione per classe (modalità multiclasse)	50
Numero minimo di documenti di formazione per classe (modalità multi-etichetta)	10
Numero minimo di documenti di formazione (modalità multietichetta)	50

Classificazione per documenti di testo semplice

Si crea e si addestra un modello di testo semplice utilizzando documenti di input in testo semplice. Amazon Comprehend fornisce operazioni asincrone e in tempo reale per classificare documenti di testo semplice utilizzando un modello di testo semplice.

Addestramento

La tabella seguente descrive le quote relative all'addestramento di un classificatore personalizzato con documenti di testo semplice.

Description	Quota/Linea guida
Dimensione totale di tutti i file del lavoro di formazione	5 GB
Numero massimo di file manifest aumentati per l'addestramento di un classificatore personalizzato	5
Numero massimo di nomi di attributi per ogni file manifest aumentato	5

Description	Quota/Linea guida
Lunghezza massima del nome dell'attributo	63 caratteri

Analisi in tempo reale (sincrona)

La tabella seguente descrive le quote relative alla classificazione in tempo reale di documenti di testo semplice.

Description	Quota/Linea guida
Numero massimo di documenti per richiesta sincrona	1
Dimensione massima del documento di testo (codifica UTF-8)	10 KB

Analisi asincrona

La tabella seguente descrive le quote relative alla classificazione asincrona di documenti di testo semplice.

Description	Quota/Linea guida
Dimensione totale di tutti i file in un processo asincrono	5 GB
Dimensione massima per un file, un documento per file	10 MB
Numero massimo di file, un documento per file	1.000.000
Numero massimo totale di righe, un documento per riga (per tutti i file richiesti)	1.000.000

Classificazione per documenti semistrutturati

Questa sezione descrive le linee guida e le quote per la classificazione dei documenti semistrutturati. Per classificare i documenti semistrutturati, utilizzate un modello di documento nativo che avete addestrato con documenti di input nativi.

Addestramento di un modello di documento nativo con documenti semistruutturati

La tabella seguente descrive le quote relative alla formazione di un classificatore personalizzato con documenti semistruutturati, come documenti PDF, documenti Word e file di immagine.

Description	Quota/Linea guida
Numero massimo di pagine in tutti i documenti	10.000
Dimensione massima del file di annotazioni (tutte le dimensioni dei file CSV combinate)	5 MB
Dimensione del corpus di documenti (documenti di formazione e test)	10 GB
Dimensioni dei file per i file di formazione e test	
Dimensioni del file di immagine (JPG, PNG, TIFF).	1 byte: 10 MB. File TIFF: massimo una pagina.
Dimensioni della pagina per i documenti PDF	1 byte — 10 MB
Dimensioni della pagina per i documenti Word	1 byte—10 MB
Dimensione JSON dell'output dell'API Amazon Textract	1 byte—1 MB

Analisi (sincrona) in tempo reale

Questa sezione descrive le quote relative alla classificazione in tempo reale dei documenti semistruutturati.

La tabella seguente mostra le dimensioni massime dei file per i documenti di input. Per tutti i tipi di documenti di input, il file di input può contenere al massimo una pagina, con non più di 10.000 caratteri.

Tipo di file	Dimensione massima (API)	Dimensione massima (console)
Documenti di testo UTF-8	10 KB	10 KB
Documenti PDF	10 MB	5 MB
Documenti Word	10 MB	5 MB
File di immagine	10 MB	5 MB
Dimensioni di output dell'API Amazon Textract	1 MB	N/A

Analisi asincrona

La tabella seguente descrive le quote relative alla classificazione asincrona dei documenti semistrutturati.

Description	Quota/Linea guida
Numero massimo di pagine in tutti i documenti di input per un lavoro	25.000
Dimensione del corpus del documento	25 GB
Dimensioni del file di immagine (JPG, PNG o TIFF)	1 byte: 10 MB. File TIFF: massimo una pagina.
Dimensioni della pagina per i documenti PDF	1 byte — 10 MB
Dimensioni della pagina per i documenti Word	1 byte—10 MB
Dimensione JSON dell'output dell'API Textract	1 byte—1 MB.

Riconoscimento personalizzato delle entità

Questa sezione descrive le linee guida e le quote per le seguenti operazioni di riconoscimento personalizzato delle entità:

- I lavori di formazione di Entity Recognizer sono iniziati con l'operazione [CreateEntityRecognizer](#).
- I processi di riconoscimento asincrono delle entità sono iniziati con l'operazione. [StartEntitiesDetectionJob](#)
- Richieste di riconoscimento sincrono di entità utilizzando l'operazione. [DetectEntities](#)

Riconoscimento personalizzato delle entità per documenti di testo semplice

Amazon Comprehend fornisce operazioni asincrone e di sincronizzazione per analizzare documenti di testo semplice con un riconoscimento di entità personalizzato.

Addestramento

Questa sezione descrive le quote relative all'addestramento di un riconoscitore di entità personalizzato per analizzare documenti di testo semplice. Per addestrare il modello, puoi fornire un elenco di entità o un set di documenti di testo annotati.

La tabella seguente descrive le quote relative all'addestramento del modello con un elenco di entità.

Description	Quota/Linea guida
Numero di entità per modello	1—25
Dimensione del documento (UTF-8)	1—5.000 byte
Numero di elementi nell'elenco delle entità	1—1 milione
Lunghezza della singola voce (dopo la striscia) nell'elenco delle voci	1—5.000
Dimensione del corpus dell'elenco di entità (tutti i documenti in testo semplice combinati)	5 KB —200 MB

La tabella seguente descrive le quote relative all'addestramento del modello con documenti di testo annotati.

Description	Quota/Linea guida
Numero di entità per identificatore di entità model/custom	1—25
Dimensione del documento (UTF-8)	1—5.000 byte
Numero di documenti (vedi Annotazioni in testo semplice)	3-200.000
Dimensione del corpus del documento (tutti i documenti in testo semplice combinati)	5 KB - 200 MB
Numero minimo di annotazioni per entità	25

Analisi (sincrona) in tempo reale

La tabella seguente descrive le quote relative all'analisi in tempo reale di documenti di testo semplice.

Description	Quota/Linea guida
Numero massimo di documenti per richiesta sincrona	1
Dimensione massima del documento di testo (codifica UTF-8)	5 KB

Analisi asincrona

La tabella seguente descrive le quote relative al riconoscimento asincrono delle entità di documenti di testo semplice.

Description	Quota/Linea guida
Dimensione del documento (UTF-8)	1 byte—1 MB
Numero massimo di file, un documento per file	1.000.000
Numero massimo totale di righe, un documento per riga (per tutti i file richiesti)	1.000.000

Description	Quota/Linea guida
Dimensione del corpus del documento (tutti i documenti in testo semplice combinati)	1 byte: 5 GB

Riconoscimento personalizzato delle entità per documenti semistruutturati

Amazon Comprehend fornisce operazioni asincrone e di sincronizzazione per analizzare documenti semistruutturati con un riconoscimento di entità personalizzato. È necessario addestrare il modello utilizzando documenti PDF annotati.

Addestramento

La tabella seguente descrive le quote relative all'addestramento di un riconoscimento di entità personalizzato (CreateEntityRecognizer) per analizzare documenti semistruutturati.

Description	Quota/Linea guida
Numero di entità per identificatore di entità model/custom	1—25
Dimensione massima del file di annotazione (UTF-8 JSON)	5 MB
Numero di documenti	250—10.000
Dimensione del corpus del documento (tutti i documenti in testo semplice combinati)	5 KB—1 GB
Numero minimo di annotazioni per entità	100
Numero massimo di file manifest aumentati per l'addestramento di un riconoscitore di entità personalizzato	5
Numero massimo di nomi di attributi per ogni file manifest aumentato	5
Lunghezza massima del nome dell'attributo	63 caratteri

Analisi in tempo reale (sincrona)

Questa sezione descrive le quote relative all'analisi in tempo reale di documenti semistrutturati.

La tabella seguente mostra le dimensioni massime dei file per i documenti di input. Per tutti i tipi di documenti di input, il file di input può contenere al massimo una pagina, con non più di 10.000 caratteri.

Tipo di file	Dimensione massima (API)	Dimensione massima (console)
Documenti di testo UTF-8	10 KB	10 KB
Documenti PDF	10 MB	5 MB
Documenti Word	10 MB	5 MB
File di immagine	10 MB	5 MB
Texttract dei file di output	1 MB	N/A

Analisi asincrona

Questa sezione descrive le quote per l'analisi asincrona di documenti semistrutturati.

Description	Quota/Linea guida
Dimensioni dell'immagine (JPG o PNG)	1 byte—10 MB
Dimensione dell'immagine (TIFF)	1 byte: 10 MB. Massimo una pagina.
Dimensioni del documento (PDF)	1 byte: 50 MB
Dimensioni del documento (Docx)	1 byte: 5 MB
Dimensioni del documento (UTF-8)	1 byte—1 MB

Description	Quota/Linea guida
Numero massimo di file, un documento per file (un documento per riga non è consentito per file di immagini o documenti) PDF/Word	500
Numero massimo di pagine per un file PDF o Docx	100
Dimensione del corpus del documento dopo l'estrazione del testo (testo semplice, tutti i file combinati)	1 byte: 5 GB

Per ulteriori informazioni sui limiti per le immagini, consulta [Hard Limits in Amazon Textract](#)

Quote per i volani

Utilizza i volani per gestire l'addestramento e il monitoraggio delle versioni dei modelli personalizzati per la classificazione personalizzata e il riconoscimento personalizzato delle entità. Per ulteriori informazioni su Flywheels, vedere. [Volani](#)

Quote generali per i volani

Le seguenti quote si applicano ai volani e alle iterazioni del volano.

Description	Quota/Linea guida
Numero massimo di volani	50
Numero massimo di volani nello stato CREATING	10
Numero massimo di set di dati di addestramento per volano	50
Numero massimo di set di dati di test per volano	50
Numero massimo di set di dati con stato INGESTING	10
Numero massimo di iterazioni del volano in corso per account	10

Quote del set di dati per modelli di classificazione personalizzati

Quando si inserisce un set di dati per un volano associato a un modello di classificazione personalizzato, si applicano le seguenti quote.

Description	Quota/linea guida
Numero minimo di documenti di formazione per classe (modalità multietichetta)	50
Numero massimo di documenti di formazione	1.000.000
Dimensione minima del set di dati	500 byte
Dimensione massima del set di dati	5 GB
Dimensione massima per un file, un documento per file	10 MB

Quote del set di dati per modelli di riconoscimento delle entità personalizzati

Quando si inserisce un set di dati per un volano associato a un modello di riconoscimento delle entità personalizzato, si applicano le seguenti quote.

Description	Quota/linea guida
Dimensione massima del documento	5 KB
Numero minimo di documenti di formazione	3
Numero massimo di documenti di formazione	200.000
Numero minimo di annotazioni per entità	25
Dimensione massima del set di dati	200 MB

Tutorial e altre risorse

Tutorial e altre risorse per Amazon Comprehend.

Argomenti

- [Tutorial: analisi delle informazioni ricavate dalle recensioni dei clienti con Amazon Comprehend](#)
- [Utilizzo dei punti di accesso Lambda a oggetti Amazon S3 per informazioni di identificazione personale \(PII\)](#)
- [Soluzione: analisi del testo con Amazon Comprehend e OpenSearch](#)

Tutorial: analisi delle informazioni ricavate dalle recensioni dei clienti con Amazon Comprehend

Questo tutorial spiega come usare Amazon Comprehend con [Amazon Simple Storage Service](#) e [Amazon Quick](#) per ottenere informazioni preziose sui tuoi documenti. [AWS Glue](#) [Amazon Athena](#) Amazon Comprehend è in grado di estrarre sentimenti (lo stato d'animo di un documento) ed entità (nomi di persone, organizzazioni, eventi, date, prodotti, luoghi, quantità e titoli) da testo non strutturato.

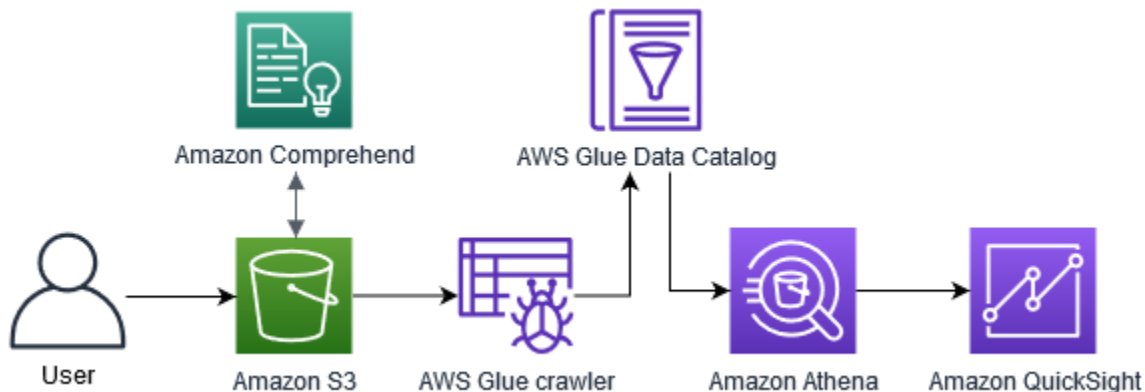
Ad esempio, puoi ottenere informazioni utili dalle recensioni dei clienti. In questo tutorial, analizzerai un set di dati di esempio di recensioni dei clienti su un romanzo. Utilizzi l'analisi del sentiment di Amazon Comprehend per determinare se i clienti si sentono positivi o negativi nei confronti del romanzo. Puoi anche utilizzare l'analisi delle entità di Amazon Comprehend per scoprire menzioni di entità importanti, come romanzi o autori correlati. Dopo aver seguito questo tutorial, potresti scoprire che oltre il 50% delle recensioni sono positive. Potresti anche scoprire che i clienti confrontano gli autori ed esprimono interesse per altri romanzi classici.

In questo tutorial, esegui le seguenti operazioni:

- Archivia un set di dati di esempio di recensioni in [Amazon Simple Storage Service](#) (Amazon S3). Amazon Simple Storage Service è un servizio di storage di oggetti.
- Usa [Amazon Comprehend](#) per analizzare il sentimento e le entità nei documenti di revisione.
- Usa un [AWS Glue](#) crawler per archiviare i risultati dell'analisi in un database. AWS Glue è un servizio di estrazione, trasformazione e caricamento (ETL) che consente di catalogare e pulire i dati per l'analisi.

- Esegui [Amazon Athena](#) query per pulire i dati. Amazon Athena è un servizio di interrogazione interattivo senza server.
- Crea visualizzazioni con i tuoi dati in [Amazon QuickSight](#). Quick è uno strumento di business intelligence senza server per estrarre informazioni dai tuoi dati.

Il diagramma seguente mostra il flusso di lavoro.



Tempo stimato per completare questo tutorial: 1 ora

Costo stimato: Alcune delle azioni di questo tutorial comportano addebiti a carico del tuo Account AWS. Per informazioni sui costi di ciascuno di questi servizi, consulta le seguenti pagine dei prezzi.

- [Prezzi di Amazon S3](#)
- [Prezzi di Amazon Comprehend](#)
- [AWS Glue prezzi](#)
- [Prezzi di Amazon Athena](#)
- [Prezzi rapidi](#)

Argomenti

- [Prerequisiti](#)
- [Fase 1: aggiunta di documenti ad Amazon S3](#)
- [Fase 2: \(solo CLI\) creazione di un ruolo IAM per Amazon Comprehend](#)
- [Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3](#)
- [Fase 4: Preparazione dell'output di Amazon Comprehend per la visualizzazione dei dati](#)
- [Fase 5: Visualizzazione dell'output di Amazon Comprehend in Quick](#)

Prerequisiti

Per completare questo tutorial, è necessario quanto segue:

- Un Account AWS. Per informazioni sulla configurazione di un Account AWS, vedere [Configurazione](#).
- Un'entità IAM (utente, gruppo o ruolo). Per scoprire come configurare utenti e gruppi per il tuo account, consulta il tutorial [introduttivo](#) nella Guida per l'utente IAM.
- La seguente politica di autorizzazione allegata al tuo utente, gruppo o ruolo. La politica concede alcune delle autorizzazioni necessarie per completare questo tutorial. Il prossimo prerequisito descrive le autorizzazioni aggiuntive necessarie.

JSON

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

Utilizza la policy precedente per creare una policy IAM e collegarla al tuo gruppo o utente. Per informazioni sulla creazione di una policy IAM, consulta [Creating IAM policies](#) nella IAM User Guide. Per informazioni su come allegare una policy IAM, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida](#) per l'utente IAM.

- Policy gestite collegate al tuo gruppo o utente IAM. Oltre alla politica precedente, devi anche allegare le seguenti politiche AWS gestite al tuo gruppo o utente:
 - AWSGlueConsoleFullAccess
 - AWSQuicksightAthenaAccess

Queste politiche gestite ti danno l'autorizzazione all'uso AWS Glue e a Quick. Amazon Athena Per informazioni su come allegare una policy IAM, consulta [Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida](#) per l'utente IAM.

Fase 1: aggiunta di documenti ad Amazon S3

Prima di iniziare i processi di analisi di Amazon Comprehend, devi archiviare un set di dati di esempio delle recensioni dei clienti in Amazon Simple Storage Service (Amazon S3). Amazon S3 ospita i dati in contenitori chiamati bucket. Amazon Comprehend è in grado di analizzare i documenti archiviati in un bucket e invia i risultati dell'analisi a un bucket. In questo passaggio, crei un bucket S3, crei cartelle di input e output nel bucket e carichi un set di dati di esempio nel bucket.

Argomenti

- [Prerequisiti](#)
- [Scarica dati di esempio](#)
- [Crea un bucket Amazon S3](#)
- [\(Solo console\) crea cartelle](#)
- [Caricate i dati di input](#)

Prerequisiti

Prima di iniziare, esamina [Tutorial: analisi delle informazioni ricavate dalle recensioni dei clienti con Amazon Comprehend](#) e completa i prerequisiti.

Scarica dati di esempio

Il seguente set di dati di esempio contiene recensioni Amazon tratte dal set di dati più ampio «Amazon reviews - Full», che è stato pubblicato con l'articolo «Character-level Convolutional Networks for Text Classification» (Xiang Zhang et al., 2015). Scarica il set di dati sul tuo computer.

Per ottenere i dati di esempio

1. Scarica il file zip [tutorial-reviews-data.zip](#) sul tuo computer.
2. Estrai il file zip sul tuo computer. Ci sono due file. Il file `THIRD_PARTY_LICENSES.txt` è la licenza open source per il set di dati pubblicato da Xiang Zhang et al. Il file `amazon-reviews.csv` è il set di dati che analizzi nel tutorial.

Crea un bucket Amazon S3

Dopo aver scaricato il set di dati di esempio, crea un bucket Amazon S3 per archiviare i dati di input e output. Puoi creare un bucket S3 utilizzando la console Amazon S3 o AWS Command Line Interface (AWS CLI).

Crea un bucket Amazon S3 (console)

Nella console Amazon S3, crei un bucket con un nome unico in tutto AWS.

Per creare un bucket S3 (console)

1. Accedi a Console di gestione AWS e apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. In Bucket, scegli Crea bucket.
3. Per Bucket name, inserisci un nome univoco globale che descriva lo scopo del bucket.
4. Per Regione, scegli la AWS regione in cui desideri creare il bucket. La regione scelta deve supportare Amazon Comprehend. Per ridurre la latenza, scegli la AWS regione più vicina alla tua posizione geografica supportata da Amazon Comprehend. Per un elenco delle regioni che supportano Amazon Comprehend, consulta la [tabella delle regioni nella](#) Global Infrastructure Guide.
5. Lascia le impostazioni predefinite per Object Ownership, Bucket per Block Public Access, Bucket Versioning e Tags.
6. Per Crittografia predefinita, scegli Disabilita.

Tip

Sebbene questo tutorial non utilizzi la crittografia, potresti volerla utilizzare per analizzare dati importanti. Per quanto riguarda end-to-end la crittografia, è possibile crittografare i dati inattivi nel bucket e anche quando si eseguono processi di analisi. Per ulteriori informazioni sulla crittografia con AWS, vedi [Cos'è? AWS Key Management Service](#) nella Guida per gli AWS Key Management Service sviluppatori.

7. Controlla le configurazioni del bucket, quindi scegli Crea bucket.

Crea un bucket Amazon S3 (AWS CLI)

Dopo aver aperto AWS CLI, esegui il `create-bucket` comando per creare un bucket che memorizzerà i dati di input e output.

Per creare un bucket Amazon S3 (AWS CLI)

1. Per creare il tuo bucket, esegui il seguente comando in AWS CLI. Sostituisci `amzn-s3-demo-bucket` con un nome per il bucket che sia unico in tutto. AWS

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket
```

Per impostazione predefinita, il comando crea un bucket nella regione `us-east-1` AWS. Per creare un bucket in un'altra regione AWS, aggiungi il `LocationConstraint` parametro per specificare la tua regione. Ad esempio, il comando seguente crea un bucket nella `us-west-2` regione.

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket  
--region us-west-2 --create-bucket-configuration LocationConstraint=us-west-2
```

Tieni presente che solo alcune regioni supportano Amazon Comprehend. Per un elenco delle regioni che supportano Amazon Comprehend, consulta la [tabella delle regioni nella](#) Global Infrastructure Guide.

2. Per assicurarti che il bucket sia stato creato correttamente, esegui il comando seguente. Il comando elenca tutti i bucket S3 associati al tuo account.

```
aws s3 ls
```

(Solo console) crea cartelle

Quindi, crea due cartelle nel tuo bucket S3. La prima cartella è per i dati di input. La seconda cartella è dove Amazon Comprehend invia i risultati dell'analisi. Se utilizzi la console Amazon S3, devi creare manualmente le cartelle. Se utilizzi AWS CLI, puoi creare cartelle quando carichi il set di dati di esempio o esegui un processo di analisi. Per questo motivo, forniamo una procedura per la creazione di cartelle solo per gli utenti della console. Se utilizzi la AWS CLI, creerai cartelle in entrata [Caricate i dati di input](#) e in uscita. [Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3](#)

Per creare cartelle nel bucket S3 (console)

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. In Bucket, scegli il tuo bucket dall'elenco dei bucket.
3. Nella scheda Panoramica, scegli Crea cartella.
4. Per il nuovo nome della cartella, inserisci input.
5. Per le impostazioni di crittografia, scegli Nessuno (Usa le impostazioni del bucket).
6. Scegli Save (Salva).
7. Ripetete i passaggi da 3 a 6 per creare un'altra cartella per l'output dei lavori di analisi, ma nel passaggio 4 inserite il nuovo nome output della cartella.

Caricate i dati di input

Ora che hai un bucket, carica il set di dati `amazon-reviews.csv` di esempio. Puoi caricare dati su bucket S3 con la console Amazon S3 o il AWS CLI

Carica documenti di esempio in un bucket (console)

Nella console Amazon S3, carica il file del set di dati di esempio nella cartella di input.

Per caricare i documenti di esempio (console)

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. In Bucket, scegli il tuo bucket dall'elenco dei bucket.
3. Scegli la **input** cartella, quindi scegli Carica.
4. Scegli Aggiungi file, quindi scegli il `amazon-reviews.csv` file sul tuo computer.
5. Lascia le altre impostazioni ai valori predefiniti.
6. Scegli Carica.

Carica documenti di esempio in un bucket (AWS CLI)

Crea una cartella di input nel tuo bucket S3 e carica il file del set di dati nella nuova cartella con il comando. `cp`

Per caricare i documenti di esempio ()AWS CLI

1. Per caricare il `amazon-reviews.csv` file in una nuova cartella nel bucket, esegui il AWS CLI comando seguente. Sostituisci `amzn-s3-demo-bucket` con il nome del tuo bucket. Aggiungendo il percorso `/input/` alla fine, Amazon S3 crea automaticamente una nuova cartella chiamata `input` nel bucket e carica il file del set di dati in quella cartella.

```
aws s3 cp amazon-reviews.csv s3://amzn-s3-demo-bucket/input/
```

2. Per assicurarti che il file sia stato caricato correttamente, esegui il seguente comando. Il comando elenca il contenuto della `input` cartella del bucket.

```
aws s3 ls s3://amzn-s3-demo-bucket/input/
```

Ora hai un bucket S3 con il `amazon-reviews.csv` file in una cartella chiamata `input`. Se hai usato la console, hai anche una `output` cartella nel bucket. Se hai utilizzato AWS CLI, creerai la cartella di `output` durante l'esecuzione dei processi di analisi di Amazon Comprehend.

Fase 2: (solo CLI) creazione di un ruolo IAM per Amazon Comprehend

Questo passaggio è necessario solo se si utilizza AWS Command Line Interface (AWS CLI) per completare questo tutorial. Se utilizzi la console Amazon Comprehend per eseguire i processi di analisi, passa a [Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3](#)

Per eseguire lavori di analisi, Amazon Comprehend richiede l'accesso al bucket Amazon S3 che contiene il set di dati di esempio e conterrà l'output dei processi. I ruoli IAM consentono di controllare le autorizzazioni dei servizi o degli utenti. AWS In questa fase, crei un ruolo IAM per Amazon Comprehend. Quindi, crei e alleggi a questo ruolo una policy basata sulle risorse che concede ad Amazon Comprehend l'accesso al tuo bucket S3. Al termine di questa fase, Amazon Comprehend disporrà delle autorizzazioni necessarie per accedere ai dati di input, archiviare l'output ed eseguire lavori di analisi del sentimento e delle entità.

Per ulteriori informazioni sull'utilizzo di IAM con Amazon Comprehend, consulta [Come funziona Amazon Comprehend con IAM](#)

Argomenti

- [Prerequisiti](#)
- [Creazione di un ruolo IAM](#)

- [Associa una policy IAM al ruolo IAM](#)

Prerequisiti

Prima di iniziare, esegui queste attività:

- Completo [Fase 1: aggiunta di documenti ad Amazon S3](#).
- Utilizza un editor di codice o di testo per salvare le policy JSON e tenere traccia dei tuoi Amazon Resource Names (ARNs).

Creazione di un ruolo IAM

Per accedere al tuo bucket Amazon Simple Storage Service (Amazon S3), Amazon Comprehend deve assumere un ruolo (IAM). AWS Identity and Access Management Il ruolo IAM dichiara Amazon Comprehend come entità affidabile. Dopo che Amazon Comprehend avrà assunto il ruolo e diventerà un'entità affidabile, puoi concedere le autorizzazioni di accesso al bucket ad Amazon Comprehend. In questo passaggio, crei un ruolo che etichetta Amazon Comprehend come entità affidabile. Puoi creare un ruolo con la AWS CLI o la console Amazon Comprehend. Per usare la console, vai a. [Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3](#)

La console Amazon Comprehend ti consente di selezionare ruoli in cui il nome del ruolo contiene «Comprehend» e la politica di attendibilità include. comprehend.amazonaws.com Configura i ruoli creati dalla CLI per soddisfare questi criteri se desideri che la console li visualizzi.

Per creare un ruolo IAM per Amazon Comprehend (CLI AWS)

1. Salva la seguente politica di attendibilità come documento JSON richiamato comprehend-trust-policy.json in un editor di codice o di testo sul tuo computer. Questa politica di fiducia dichiara Amazon Comprehend come entità affidabile e le consente di assumere un ruolo IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```
"Effect": "Allow",
"Principal": {
  "Service": "comprehend.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
```

2. Per creare il ruolo IAM, esegui il comando seguente AWS CLI . Il comando crea un ruolo IAM chiamato `AmazonComprehendServiceRole-access-role` e associa la policy di fiducia al ruolo. Sostituiscilo *path/* con il percorso del documento JSON del tuo computer locale.

```
aws iam create-role --role-name AmazonComprehendServiceRole-access-role
--assume-role-policy-document file:///path/comprehend-trust-policy.json
```

Tip

Se viene visualizzato un messaggio di errore nell'analisi dei parametri, il percorso del file dei criteri di attendibilità JSON è probabilmente errato. Fornisci il percorso relativo del file in base alla tua home directory.

3. Copia l'Amazon Resource Name (ARN) e salvalo in un editor di testo. L'ARN ha un formato simile a. *arn:aws:iam::123456789012:role/AmazonComprehendServiceRole-access-role* È necessario questo ARN per eseguire i processi di analisi di Amazon Comprehend.

Associa una policy IAM al ruolo IAM

Per accedere al tuo bucket Amazon S3, Amazon Comprehend necessita delle autorizzazioni per elencare, leggere e scrivere. Per concedere ad Amazon Comprehend le autorizzazioni richieste, crea e allega una policy IAM al tuo ruolo IAM. La policy IAM consente ad Amazon Comprehend di recuperare i dati di input dal bucket e scrivere i risultati delle analisi nel bucket. Dopo aver creato la policy, la colleghi al tuo ruolo IAM.

Per creare una policy IAM (AWS CLI)

1. Salva la seguente politica localmente come documento JSON chiamato `comprehend-access-policy.json`. Concede ad Amazon Comprehend l'accesso al bucket S3 specificato.

JSON

```
{
  "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. Per creare la policy di accesso al bucket S3, esegui il comando seguente. AWS CLI *path*/Sostituiscilo con il percorso del documento JSON del tuo computer locale.

```
aws iam create-policy --policy-name comprehend-access-policy
--policy-document file://path/comprehend-access-policy.json
```

3. Copia l'ARN della politica di accesso e salvalo in un editor di testo. L'ARN ha un formato simile a. `arn:aws:iam::123456789012:policy/comprehend-access-policy` Questo ARN ti serve per collegare la tua policy di accesso al tuo ruolo IAM.

Per collegare la policy IAM al tuo ruolo IAM ()AWS CLI

- Eseguire il seguente comando seguente. Sostituisci `policy-arn` con l'ARN della politica di accesso che hai copiato nel passaggio precedente.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name AmazonComprehendServiceRole-access-role
```

Ora disponi di un ruolo IAM chiamato `AmazonComprehendServiceRole-access-role` che prevede una policy di fiducia per Amazon Comprehend e una policy di accesso che concede ad Amazon Comprehend l'accesso al tuo bucket S3. Hai anche copiato l'ARN per il ruolo IAM in un editor di testo.

Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3

Dopo aver archiviato i dati in Amazon S3, puoi iniziare a eseguire i job di analisi di Amazon Comprehend. Un lavoro di analisi del sentiment determina lo stato d'animo generale di un documento (positivo, negativo, neutro o misto). Un processo di analisi delle entità estrae i nomi degli oggetti del mondo reale da un documento. Questi oggetti includono persone, luoghi, titoli, eventi, date, quantità, prodotti e organizzazioni. In questa fase, esegui due processi di analisi di Amazon Comprehend per estrarre il sentimento e le entità dal set di dati campione.

Argomenti

- [Prerequisiti](#)
- [Analizza il sentimento e le entità](#)

Prerequisiti

Prima di iniziare, esegui queste attività:

- Completo [Fase 1: aggiunta di documenti ad Amazon S3](#).
- (Facoltativo) Se utilizzi l'ARN AWS CLI, completa [Fase 2: \(solo CLI\) creazione di un ruolo IAM per Amazon Comprehend](#) e tieni pronto il tuo ruolo IAM.

Analizza il sentimento e le entità

Il primo lavoro che esegui analizza il sentimento di ogni recensione del cliente nel set di dati di esempio. Il secondo processo estrae le entità presenti in ogni recensione del cliente. Puoi eseguire lavori di analisi di Amazon Comprehend utilizzando la console Amazon Comprehend o il AWS CLI.

Tip

Assicurati di trovarti in una AWS regione che supporta Amazon Comprehend. Per ulteriori informazioni, consulta la [tabella delle regioni nella](#) Global Infrastructure Guide.

Analizza sentimenti ed entità (console)

Quando usi la console Amazon Comprehend, crei un lavoro alla volta. È necessario ripetere i passaggi seguenti per eseguire sia un lavoro di analisi del sentimento che quello delle entità. Tieni presente che per il primo lavoro crei un ruolo IAM, ma per il secondo lavoro puoi riutilizzare il ruolo IAM del primo lavoro. Puoi riutilizzare il ruolo IAM purché utilizzi lo stesso bucket e le stesse cartelle S3.

Per eseguire processi di analisi del sentimento e delle entità (console)

1. Assicurati di trovarti nella stessa regione in cui hai creato il bucket Amazon Simple Storage Service (Amazon S3). Se ti trovi in un'altra regione, nella barra di navigazione, scegli la AWS regione in cui hai creato il bucket S3 dal selettore Regione.
2. Apri la console Amazon Comprehend all'indirizzo <https://console.aws.amazon.com/comprehend/>
3. Scegli Launch Amazon Comprehend.
4. Nel riquadro di navigazione, scegli Analysis jobs.
5. Scegli Crea processo.
6. Nella sezione Job settings, effettuate le seguenti operazioni:
 - a. In Nome, inserisci `reviews-sentiment-analysis`.
 - b. Per Tipo di analisi, scegli Sentiment.
 - c. Per Lingua, scegli l'inglese.
 - d. Lasciate l'impostazione di crittografia Job disattivata.
7. Nella sezione Dati di input, effettuate le seguenti operazioni:

- a. Per Origine dati, scegli I miei documenti.
 - b. Per la posizione S3, scegli Browse S3, quindi scegli il tuo bucket dall'elenco dei bucket.
 - c. Nel tuo bucket S3, per Oggetti, scegli la tua cartella. **input**
 - d. Nella **input** cartella, scegli il set di dati **amazon-reviews.csv** di esempio, quindi scegli Scegli.
 - e. Per Formato di input, scegliete Un documento per riga.
8. Nella sezione Dati di output, procedi come segue:
- a. Per la posizione S3, scegli Browse S3, quindi scegli il tuo bucket dall'elenco dei bucket.
 - b. Nel tuo bucket S3, per Oggetti, scegli la cartella, quindi scegli Scegli. **output**
 - c. Lascia la crittografia disattivata.
9. Nella sezione Autorizzazioni di accesso, procedi come segue:
- a. Per il ruolo IAM, scegli Crea un ruolo IAM.
 - b. Per le autorizzazioni di accesso, scegli i bucket S3 di input e output.
 - c. Per il suffisso del nome, inserisci. **comprehend-access-role** Questo ruolo fornisce l'accesso al tuo bucket Amazon S3.
10. Scegli Crea processo.
11. Ripeti i passaggi da 1 a 10 per creare un processo di analisi delle entità. Apporta le seguenti modifiche:
- a. Nelle impostazioni Job, per Nome, immettete **reviews-entities-analysis**.
 - b. Nelle impostazioni Job, per Tipo di analisi, scegli Entità.
 - c. In Autorizzazioni di accesso, scegli Usa un ruolo IAM esistente. Per Nome del ruolo, scegli **AmazonComprehendServiceRole-comprehend-access-role** (è lo stesso ruolo che hai creato per il lavoro Sentiment).

Analizza i sentimenti e le entità (AWS CLI)

Si utilizzano i **start-entities-detection-job** comandi **start-sentiment-detection-job** and per eseguire lavori di analisi dei sentimenti e delle entità. Dopo aver eseguito ogni comando, AWS CLI viene visualizzato un oggetto JSON con un **JobId** valore che consente di accedere ai dettagli sul processo, inclusa la posizione di output S3.

Per eseguire lavori di analisi del sentimento e delle entità (AWS CLI)

1. Avvia un processo di analisi del sentiment eseguendo il seguente comando in AWS CLI. Sostituisci *arn:aws:iam::123456789012:role/comprehend-access-role* con l'ARN del ruolo IAM che hai precedentemente copiato in un editor di testo. Se la tua AWS CLI regione predefinita è diversa dalla regione in cui hai creato il bucket Amazon S3, includi il parametro e *us-east-1* sostituiscilo con `--region` la regione in cui risiede il bucket.

```
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. Dopo aver inviato il lavoro, copialo JobId e salvalo in un editor di testo. È necessario JobId trovare i file di output dal lavoro di analisi.
3. Avviate un processo di analisi delle entità eseguendo il comando seguente.

```
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. Dopo aver inviato il lavoro, copiatelo JobId e salvatelo in un editor di testo.
5. Controlla lo stato delle tue offerte di lavoro. Puoi visualizzare lo stato di avanzamento di un lavoro monitorandolo JobId.

Per tenere traccia dello stato di avanzamento del tuo lavoro di analisi dei sentimenti, esegui il comando seguente. *sentiment-job-id* Sostituiscilo con JobId quello che hai copiato dopo aver eseguito l'analisi del sentiment.

```
aws comprehend describe-sentiment-detection-job
--job-id sentiment-job-id
```

Per tenere traccia del processo di analisi delle entità, esegui il comando seguente. Sostituiscilo *entities-job-id* con JobId quello che hai copiato dopo aver eseguito l'analisi delle entità.

```
aws comprehend describe-entities-detection-job  
--job-id entities-job-id
```

Occorrono alcuni minuti prima che JobStatus venga visualizzato come COMPLETED.

Hai completato i lavori di analisi del sentimento e delle entità. Entrambi i lavori devono essere completati prima di passare alla fase successiva. Il completamento dei lavori può richiedere diversi minuti.

Fase 4: Preparazione dell'output di Amazon Comprehend per la visualizzazione dei dati

Per preparare i risultati dei lavori di analisi del sentimento e delle entità per la creazione di visualizzazioni di dati, usi e. AWS Glue Amazon Athena In questo passaggio, estrai i file dei risultati di Amazon Comprehend. Quindi, crei un AWS Glue crawler che esplora i tuoi dati e li cataloga automaticamente in tabelle in. AWS Glue Data Catalog Dopodiché, accedi e trasformi queste tabelle utilizzando un servizio di query Amazon Athena interattivo e senza server. Al termine di questo passaggio, i risultati di Amazon Comprehend sono puliti e pronti per la visualizzazione.

Per un processo di rilevamento di entità PII, il file di output è in testo semplice, non è un archivio compresso. Il nome del file di output è lo stesso del file di input e viene .out aggiunto alla fine. Non è necessario il passaggio di estrazione del file di output. Passa a [caricare i dati in un](#). AWS Glue Data Catalog

Argomenti

- [Prerequisiti](#)
- [Scarica l'output](#)
- [Estrai i file di output](#)
- [Carica i file estratti](#)
- [Carica i dati in un AWS Glue Data Catalog](#)
- [Prepara i dati per l'analisi](#)

Prerequisiti

Prima di iniziare, completa [Fase 3: Esecuzione di processi di analisi su documenti in Amazon S3](#).

Scarica l'output

Amazon Comprehend utilizza la compressione Gzip per comprimere i file di output e salvarli come archivio tar. Il modo più semplice per estrarre i file di output è scaricare gli archivi localmente.

`output.tar.gz`

In questo passaggio, scarichi gli archivi di output dei sentimenti e delle entità.

Scarica i file di output (console)

Per trovare i file di output per ogni processo, torna al lavoro di analisi nella console Amazon Comprehend. Il processo di analisi fornisce la posizione S3 per l'output, da cui è possibile scaricare il file di output.

Per scaricare i file di output (console)

1. Nella [console Amazon Comprehend](#), nel riquadro di navigazione, torna a Analysis jobs.
2. Scegli il tuo lavoro di analisi del sentiment. `reviews-sentiment-analysis`
3. In Output, scegli il link visualizzato accanto a Posizione dei dati di output. Questo ti reindirizza all'`output.tar.gz` archivio nel tuo bucket S3.
4. Nella scheda Panoramica, scegli Scarica.
5. Sul tuo computer, rinomina l'archivio come `sentiment-output.tar.gz`. Poiché tutti i file di output hanno lo stesso nome, questo ti aiuta a tenere traccia dei file Sentiment ed Entities.
6. Ripeti i passaggi 1-4 per trovare e scaricare l'output del tuo `reviews-entities-analysis` lavoro. Sul tuo computer, rinomina l'archivio come `entities-output.tar.gz`

Scarica i file di output (AWS CLI)

Per trovare i file di output per ogni lavoro, utilizzate il `JobId` comando `from the analysis` per trovare la posizione S3 dell'output. Quindi, utilizzate il `cp` comando per scaricare il file di output sul computer.

Per scaricare i file di output (AWS CLI)

1. Per elencare i dettagli sul tuo lavoro di analisi del sentiment, esegui il comando seguente.
`sentiment-job-id` Sostituiscilo con il sentimento JobId che hai salvato.


```
aws comprehend describe-sentiment-detection-job --job-id sentiment-job-id
```

Se hai perso di vista il tuo JobId profilo, puoi eseguire il seguente comando per elencare tutti i tuoi lavori sentimentali e filtrarli per nome.

```
aws comprehend list-sentiment-detection-jobs  
--filter JobName="reviews-sentiment-analysis"
```

2. Nell'OutputDataConfigoggetto, trova il S3Uri valore. Il S3Uri valore dovrebbe essere simile al formato seguente:*s3://amzn-s3-demo-bucket/.../output/output.tar.gz*. Copia questo valore in un editor di testo.
3. Per scaricare l'archivio dei risultati dei sentimenti nella tua directory locale, esegui il seguente comando. Sostituisci il percorso del bucket S3 con S3Uri quello copiato nel passaggio precedente. *path*/Sostituiscilo con il percorso della cartella della tua directory locale. Il nome *sentiment-output.tar.gz* sostituisce il nome dell'archivio originale per aiutarti a tenere traccia dei file di sentiment ed entities.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz  
path/sentiment-output.tar.gz
```

4. Per elencare i dettagli sul processo di analisi delle entità, esegui il comando seguente.

```
aws comprehend describe-entities-detection-job  
--job-id entities-job-id
```

Se non conosci il tuoJobId, esegui il comando seguente per elencare tutti i lavori delle entità e filtrare il lavoro per nome.

```
aws comprehend list-entities-detection-jobs  
--filter JobName="reviews-entities-analysis"
```

5. Dall'OutputDataConfigoggetto nella descrizione del lavoro della tua entità, copia il S3Uri valore.
6. Per scaricare l'archivio di output delle entità nella directory locale, esegui il comando seguente. Sostituisci il percorso del bucket S3 con S3Uri quello copiato nel passaggio precedente. *path*/Sostituiscilo con il percorso della cartella della tua directory locale. Il nome *entities-output.tar.gz* sostituisce il nome dell'archivio originale.

```
aws s3 cp s3://amzn-s3-demo-bucket/.../output/output.tar.gz  
path/entities-output.tar.gz
```

Estrai i file di output

Prima di poter accedere ai risultati di Amazon Comprehend, decomprimi gli archivi dei sentimenti e delle entità. Puoi utilizzare il file system locale o un terminale per decomprimere gli archivi.

Estrarre i file di output (file system GUI)

Se usi macOS, fai doppio clic sull'archivio nel tuo file system GUI per estrarre il file di output dall'archivio.

Se usi Windows, puoi usare uno strumento di terze parti come 7-Zip per estrarre i file di output dal tuo file system GUI. In Windows, è necessario eseguire due passaggi per accedere al file di output nell'archivio. Prima decomprimi l'archivio, quindi estrai l'archivio.

Rinomina il file dei sentimenti come `sentiment-output` e il file delle entità come `entities-output` per distinguere i file di output.

Estrai i file di output (terminale)

Se usi Linux o macOS, puoi usare il tuo terminale standard. Se usi Windows, devi avere accesso a un ambiente in stile Unix, come Cygwin, per eseguire i comandi tar.

Per estrarre il file di output dei sentimenti dall'archivio dei sentimenti, esegui il seguente comando nel tuo terminale locale.

```
tar -xvf sentiment-output.tar.gz --transform 's,^,sentiment-,'
```

Nota che il `--transform` parametro aggiunge il prefisso `sentiment-` al file di output all'interno dell'archivio, rinominando il file come `sentiment-output`. Ciò consente di distinguere tra i file di output di sentiment e quelli delle entità e di evitare la sovrascrittura.

Per estrarre il file di output delle entità dall'archivio delle entità, esegui il seguente comando nel tuo terminale locale.

```
tar -xvf entities-output.tar.gz --transform 's,^,entities-,'
```

Il `--transform` parametro aggiunge il prefisso `entities-` al nome del file di output.

 Tip

Per risparmiare sui costi di archiviazione in Amazon S3, puoi comprimere nuovamente i file con Gzip prima di caricarli. È importante decomprimere e decomprimere gli archivi originali perché non è AWS Glue possibile leggere automaticamente i dati da un archivio tar. Tuttavia, AWS Glue può leggere da file in formato Gzip.

Carica i file estratti

Dopo aver estratto i file, caricali nel tuo bucket. È necessario archiviare i file di output dei sentimenti e delle entità in cartelle separate AWS Glue per leggere correttamente i dati. Nel tuo bucket, crea una cartella per i risultati di sentiment estratti e una seconda cartella per i risultati delle entità estratte. Puoi creare cartelle con la console Amazon S3 o il AWS CLI

Carica i file estratti su Amazon S3 (console)

Nel tuo bucket S3, crea una cartella per il file dei risultati del sentiment estratto e una cartella per il file dei risultati delle entità. Quindi, carica i file dei risultati estratti nelle rispettive cartelle.

Per caricare i file estratti su Amazon S3 (console)

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. In Bucket, scegli il tuo bucket, quindi scegli Crea cartella.
3. Per il nuovo nome della cartella, inserisci **sentiment-results** e scegli Salva. Questa cartella conterrà il file di output del sentiment estratto.
4. Nella scheda Panoramica del bucket, dall'elenco dei contenuti del bucket, scegli la nuova cartella. **sentiment-results** Scegli Carica.
5. Scegli Aggiungi file, scegli il **sentiment-output** file dal tuo computer locale, quindi scegli Avanti.
6. Lascia le opzioni per Gestisci utenti, Accedi per altri Account AWS e Gestisci le autorizzazioni pubbliche come impostazioni predefinite. Scegli Next (Successivo).
7. Per la classe Storage, scegli Standard. Lascia le opzioni per Crittografia, Metadati e Tag come predefinite. Scegli Next (Successivo).
8. Controlla le opzioni di caricamento, quindi scegli Carica.
9. Ripeti i passaggi 1-8 per creare una cartella denominata **entities-results** e caricarvi il **entities-output** file.

Carica i file estratti su Amazon AWS CLI S3 ()

Puoi creare una cartella nel tuo bucket S3 durante il caricamento di un file con il comando. cp

Per caricare i file estratti su Amazon AWS CLI S3 ()

1. Crea una cartella sentiment e carica il tuo file di sentiment eseguendo il seguente comando. Sostituisci *path/* con il percorso locale del file di output dei sentimenti estratto.

```
aws s3 cp path/sentiment-output s3://amzn-s3-demo-bucket/sentiment-results/
```

2. Crea una cartella di output delle entità e carica il file delle entità eseguendo il seguente comando. Sostituisci *path/* con il percorso locale del file di output delle entità estratte.

```
aws s3 cp path/entities-output s3://amzn-s3-demo-bucket/entities-results/
```

Carica i dati in un AWS Glue Data Catalog

Per inserire i risultati in un database, puoi usare un AWS Glue crawler. Un AWS Glue crawler analizza i file e scopre lo schema dei dati. Quindi organizza i dati in tabelle in un AWS Glue Data Catalog (un database senza server). È possibile creare un crawler con la console o il AWS Glue . AWS CLI

Carica i dati in una AWS Glue Data Catalog (console)

Crea un AWS Glue crawler che scansioni le tue cartelle `sentiment-results` e `entities-results` quelle delle cartelle separatamente. Un nuovo ruolo IAM per AWS Glue concede al crawler l'autorizzazione ad accedere al tuo bucket S3. Crei questo ruolo IAM durante la configurazione del crawler.

Per caricare i dati in una AWS Glue Data Catalog (console)

1. Assicurati di trovarti in una regione che supporti AWS Glue. Se ti trovi in un'altra regione, nella barra di navigazione, scegli una regione supportata dal selettore Regione. Per un elenco delle regioni che lo supportano AWS Glue, consulta la [tabella delle regioni](#) nella Guida globale all'infrastruttura.
2. Apri la AWS Glue console all'indirizzo <https://console.aws.amazon.com/glue/>.
3. Nel riquadro di navigazione, scegli Crawler, quindi scegli Aggiungi crawler.

4. Per il nome del crawler, inserisci e quindi scegli Avanti. **comprehend-analysis-crawler**
5. Per il tipo di sorgente Crawler, scegli Archivi dati, quindi scegli Avanti.
6. Per Aggiungi un data store, procedi come segue:
 - a. Per Scegli un datastore, scegliere S3.
 - b. Lascia vuoto il campo Connessione.
 - c. Per Scansionare i dati, scegli Percorso specificato nel mio account.
 - d. Per Includi percorso, inserisci il percorso S3 completo della cartella di output dei sentimenti:
`s3://amzn-s3-demo-bucket/sentiment-results`
 - e. Scegli Next (Successivo).
7. Per Aggiungi un altro archivio dati, scegli Sì, quindi scegli Avanti. Ripeti il passaggio 6, ma inserisci il percorso S3 completo della cartella di output delle entità:`s3://amzn-s3-demo-bucket/entities-results`.
8. Per Aggiungi un altro archivio dati, scegli No, quindi scegli Avanti.
9. Per Scegli un ruolo IAM, procedi come segue:
 - a. Scegli Crea un ruolo IAM.
 - b. Per il ruolo IAM, inserisci `glue-access-role` e quindi scegli Avanti.
10. Per Crea una pianificazione per questo crawler, scegli Esegui su richiesta e scegli Avanti.
11. Per Configura l'output del crawler, procedi come segue:
 - a. Per Database, scegliete Aggiungi database.
 - b. Per Database name (Nome database), immettere `comprehend-results`. Questo database memorizzerà le tabelle di output di Amazon Comprehend.
 - c. Lascia le altre opzioni sulle impostazioni predefinite e scegli Avanti.
12. Controlla le informazioni del crawler, quindi scegli Fine.
13. Nella console Glue, in Crawlers, scegli `comprehend-analysis-crawler` e scegli Run crawler. Il completamento del crawler può richiedere alcuni minuti.

Carica i dati in un AWS Glue Data Catalog file ()AWS CLI

Crea un ruolo IAM AWS Glue che fornisca l'autorizzazione ad accedere al tuo bucket S3. Quindi, crea un database in AWS Glue Data Catalog Infine, crea ed esegui un crawler che carichi i dati nelle tabelle del database.

Per caricare i dati in un AWS Glue Data Catalog file ()AWS CLI

1. Per creare un ruolo IAM per AWS Glue, procedi come segue:
 - a. Salva la seguente politica di attendibilità come documento JSON richiamato `glue-trust-policy.json` sul tuo computer.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- b. Per creare un ruolo IAM, esegui il comando seguente. Sostituiscilo *path/* con il percorso del documento JSON del tuo computer locale.

```
aws iam create-role --role-name glue-access-role
--assume-role-policy-document file:///path/glue-trust-policy.json
```

- c. Quando AWS CLI elenca l'Amazon Resource Number (ARN) per il nuovo ruolo, copialo e salvalo in un editor di testo.
 - d. Salva la seguente policy IAM come documento JSON richiamato `glue-access-policy.json` sul tuo computer. La policy concede il AWS Glue permesso di eseguire la scansione delle cartelle dei risultati.

JSON

```
{
  "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. Per creare la policy IAM, esegui il comando seguente. Sostituiscilo *path/* con il percorso del documento JSON del tuo computer locale.

```
aws iam create-policy --policy-name glue-access-policy
--policy-document file:///path/glue-access-policy.json
```

- f. Quando AWS CLI elenca l'ARN della politica di accesso, copialo e salvalo in un editor di testo.
- g. Associa la nuova policy al ruolo IAM eseguendo il comando seguente. Sostituisci *policy-arn* con l'ARN della policy IAM che hai copiato nel passaggio precedente.

```
aws iam attach-role-policy --policy-arn policy-arn
--role-name glue-access-role
```

- h. Allega la policy AWS gestita AWSGlueServiceRole al tuo ruolo IAM eseguendo il comando seguente.

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole
--role-name glue-access-role
```

2. Crea un AWS Glue database eseguendo il comando seguente.

```
aws glue create-database
--database-input Name="comprehend-results"
```

3. Crea un nuovo AWS Glue crawler eseguendo il comando seguente. Sostituisci *glue-iam-role-arn* con l'ARN del tuo ruolo AWS Glue IAM.

```
aws glue create-crawler
--name comprehend-analysis-crawler
--role glue-iam-role-arn
--targets S3Targets=[
{Path="s3://amzn-s3-demo-bucket/sentiment-results"},
{Path="s3://amzn-s3-demo-bucket/entities-results"}]
--database-name comprehend-results
```

4. Avvia il crawler eseguendo il comando seguente.

```
aws glue start-crawler --name comprehend-analysis-crawler
```

Il completamento del crawler può richiedere alcuni minuti.

Prepara i dati per l'analisi

Ora hai un database popolato con i risultati di Amazon Comprehend. Tuttavia, i risultati sono annidati. Per disinserirli, si eseguono alcune istruzioni SQL. Amazon Athena è un servizio di interrogazione interattivo che semplifica l'analisi dei dati in Amazon S3 utilizzando SQL standard. Athena è serverless, quindi non richiede alcuna infrastruttura da gestire e ha un pay-per-query modello di prezzo. In questo passaggio, crei nuove tabelle di dati puliti che puoi utilizzare per l'analisi e la visualizzazione. Si utilizza la console Athena per preparare i dati.

Per preparare i dati

1. Apri la console Athena all'indirizzo <https://console.aws.amazon.com/athena/>.
2. Nell'editor di query, scegli Impostazioni, quindi seleziona Gestisci.
3. Per Ubicazione dei risultati dell'interrogazione, immetters3://amzn-s3-demo-bucket/query-results/. In questo modo viene creata una nuova cartella denominata query-results nel bucket che memorizza l'output delle Amazon Athena query eseguite. Scegli Save (Salva).
4. Nell'editor delle query, scegli Editor.
5. Per Database, scegli il AWS Glue database comprehend-results che hai creato.
6. Nella sezione Tabelle, dovresti avere due tabelle chiamate sentiment_results e entities_results. Visualizzate l'anteprima delle tabelle per assicurarvi che il crawler abbia caricato i dati. Nelle opzioni di ogni tabella (i tre punti accanto al nome della tabella), scegliete

Anteprima tabella. Una breve query viene eseguita automaticamente. Controlla il riquadro Risultati per assicurarti che le tabelle contengano dati.

 Tip

Se le tabelle non contengono dati, prova a controllare le cartelle nel tuo bucket S3. Assicurati che ci sia una cartella per i risultati delle entità e una cartella per i risultati del sentiment. Quindi, prova a eseguire un nuovo AWS Glue crawler.

7. Per rimuovere la **sentiment_results** tabella, inserisci la seguente query nell'editor di query e scegli Esegui.

```
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. Per iniziare a disnidificare la tabella delle entità, inserisci la seguente query nell'editor di query e scegli Esegui.

```
CREATE TABLE entities_results_1 AS
SELECT file, line, nested FROM entities_results
CROSS JOIN UNNEST(entities) as t(nested)
```

9. Per completare l'eliminazione della tabella delle entità, inserisci la seguente query nell'editor di query e scegli Esegui query.

```
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
```

La **sentiment_results_final** tabella dovrebbe avere l'aspetto seguente, con colonne denominate file, line, sentiment, mixed, negative, neutral e positive. La tabella deve avere un valore per cella. La colonna sul sentimento descrive il sentimento generale più probabile di una particolare recensione. Le colonne miste, negative, neutre e positive forniscono punteggi per ogni tipo di 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	6.5368581090297084E-6	0.0013866390800103545	0.007405391428619623	0.9912014007568
4	amazon-reviews.csv	13	POSITIVE	4.7155481297522783E-4	0.24615342915058136	0.017713148146867752	0.7356618046760
5	amazon-reviews.csv	14	POSITIVE	1.5821871784282848E-5	0.06828905642032623	0.014075091108679771	0.9176200628280
6	amazon-reviews.csv	16	MIXED	0.9864791035652161	8.548551704734564E-4	1.0789262159960344E-4	0.0125581491738
7	amazon-reviews.csv	20	NEGATIVE	1.1621621524682269E-4	0.9815887212753296	0.004688907880336046	0.0136061981320
8	amazon-reviews.csv	21	POSITIVE	4.663573781726882E-5	0.009533549658954144	0.0015825830632820725	0.9888372421264
9	amazon-reviews.csv	23	POSITIVE	1.7699007003102452E-4	0.40269607305526733	0.0018250439316034317	0.5953019261360
10	amazon-reviews.csv	25	POSITIVE	1.8434448065818287E-6	1.15832663141191E-4	0.0010993879986926913	0.9987829327583

La **entities_results_final** tabella dovrebbe essere simile alla seguente, con colonne denominate file, line, beginoffset, endoffset, score, entity e category. La tabella deve avere un valore per cella. La colonna del punteggio indica la fiducia di Amazon Comprehend nell'entità rilevata. La categoria indica il tipo di entità rilevata da Comprehend.

Results  							
	file ▾	line ▾	beginoffset ▾	endoffset ▾	score ▾	entity ▾	category ▾
1	amazon-reviews.csv	0	15	22	0.9885989378545348	English	OTHER
2	amazon-reviews.csv	2	24	28	0.9699371997593782	2 me	QUANTITY
3	amazon-reviews.csv	2	94	95	0.6523066984191679	2	QUANTITY
4	amazon-reviews.csv	2	125	126	0.713791396412543	2	QUANTITY
5	amazon-reviews.csv	4	30	36	0.9957169942979278	kindle	COMMERCIAL_ITEM
6	amazon-reviews.csv	5	1	10	0.9979111763962706	Hawthorne	PERSON
7	amazon-reviews.csv	5	135	142	0.5065408081314243	Puritan	OTHER
8	amazon-reviews.csv	5	143	148	0.7702269458801602	Salem	LOCATION
9	amazon-reviews.csv	5	211	229	0.999675563687763	The Scarlet Letter	TITLE
10	amazon-reviews.csv	5	233	236	0.8944631322676461	one	QUANTITY

Ora che hai i risultati di Amazon Comprehend caricati nelle tabelle, puoi visualizzare ed estrarre informazioni significative dai dati.

Fase 5: Visualizzazione dell'output di Amazon Comprehend in Quick

Dopo aver archiviato i risultati di Amazon Comprehend in tabelle, puoi connetterti e visualizzare i dati con Quick. Quick è uno strumento di business intelligence (BI) AWS gestito per la visualizzazione dei dati. Quick semplifica la connessione alla fonte di dati e la creazione di immagini potenti. In questa fase, connetti Quick ai tuoi dati, crei visualizzazioni che estraggono approfondimenti dai dati e pubblichi una dashboard di visualizzazioni.

Argomenti

- [Prerequisiti](#)
- [Offri un accesso rapido](#)
- [Importa i set di dati](#)
- [Crea una visualizzazione del sentimento](#)
- [Crea una visualizzazione delle entità](#)
- [Pubblica una dashboard](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare, completa [Fase 4: Preparazione dell'output di Amazon Comprehend per la visualizzazione dei dati](#).

Offri un accesso rapido

Per importare i dati, Quick richiede l'accesso al bucket e alle tabelle Amazon Simple Storage Service (Amazon S3). Amazon Athena Per consentire l'accesso rapido ai tuoi dati, devi avere effettuato l'accesso come QuickSight amministratore e avere accesso per modificare le autorizzazioni delle risorse. Se non riesci a completare i seguenti passaggi, consulta i prerequisiti IAM dalla pagina di panoramica. [Tutorial: analisi delle informazioni ricavate dalle recensioni dei clienti con Amazon Comprehend](#)

Per consentire un accesso rapido ai tuoi dati

1. Apri la [console Quick](#).
2. Se è la prima volta che usi Quick, la console ti chiede di creare un nuovo utente amministratore fornendo un indirizzo e-mail. Per Indirizzo e-mail, inserisci lo stesso indirizzo e-mail del tuo Account AWS Scegli Continua.
3. Dopo aver effettuato l'accesso, scegli il nome del tuo profilo nella barra di navigazione e scegli Gestisci QuickSight. Devi aver effettuato l'accesso come amministratore per visualizzare l'QuickSightopzione Gestisci.
4. Scegli Sicurezza e autorizzazioni.
5. Per QuickSight accedere ai AWS servizi, scegli Aggiungi o rimuovi.
6. Seleziona Amazon S3.
7. Da Select Amazon S3 bucket, scegli il tuo bucket S3 per le autorizzazioni S3 Bucket e Write for Athena Workgroup.
8. Scegli Fine.
9. Scegliere Aggiorna.

Importa i set di dati

Prima di creare visualizzazioni, è necessario aggiungere i set di dati di sentiment ed entities a Quick. Puoi farlo con la console Quick. Puoi importare le tue tabelle unnested sentiment e unnested entities da Amazon Athena

Per importare i tuoi set di dati

1. Apri la [console Quick](#).
2. Nella barra di navigazione, in Set di dati, scegli Nuovo set di dati.
3. Per Crea un set di dati, scegli Athena.
4. Per Nome dell'origine dati, inserisci `reviews-sentiment-analysis` e scegli Crea origine dati.
5. Per Database (Database), scegli il database `comprehend-results`.
6. Per Tabelle, scegli la tabella dei sentimenti, `sentiment_results_final` quindi scegli Seleziona.
7. Scegli Importa in SPICE per un'analisi più rapida e scegli Visualize. SPICE è il motore QuickSight di calcolo in memoria che fornisce analisi più rapide rispetto all'interrogazione diretta durante la creazione di visualizzazioni.
8. Torna alla console Quick e scegli Datasets. Ripeti i passaggi 1-7 per creare un set di dati di entità, ma apporta le seguenti modifiche:
 - a. Per Nome dell'origine dati, immettere `reviews-entities-analysis`
 - b. Per Tabelle, scegli la tabella delle entità `entities_results_final`.

Crea una visualizzazione del sentimento

Ora che puoi accedere ai tuoi dati in Quick, puoi iniziare a creare visualizzazioni. Crei un grafico a torta con i dati sul sentiment di Amazon Comprehend. Il grafico a torta mostra la percentuale di recensioni positive, neutre, miste e negative.

Per visualizzare i dati sul sentiment

1. Nella console Quick, scegli Analisi, quindi scegli Nuova analisi.
2. Da I tuoi set di dati, scegli il set di dati **sentiment_results_final** sul sentiment, quindi scegli Crea analisi.
3. Nell'editor visuale, nell'elenco Campi, scegli Sentiment.

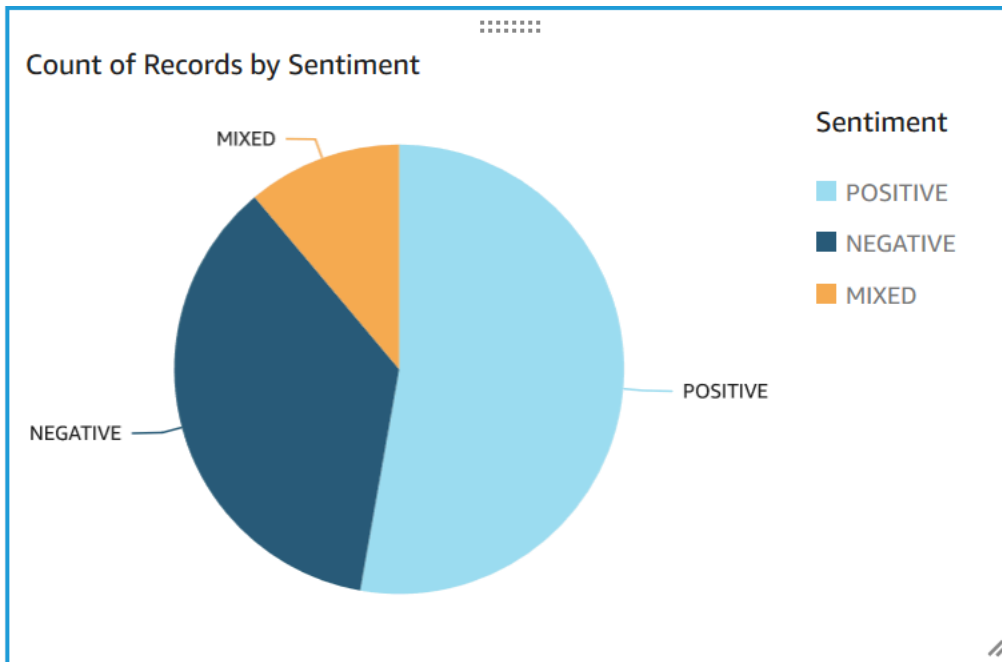
Note

I valori nell'elenco Campi dipendono dai nomi delle colonne in Amazon Athena cui hai creato le tabelle. Se sono stati modificati i nomi delle colonne forniti nelle query SQL,

i nomi degli elenchi dei campi saranno diversi dai nomi utilizzati in questi esempi di visualizzazione.

4. Per i tipi visivi, scegli Pie chart.

Viene visualizzato un grafico a torta simile al seguente con sezioni positive, neutre, miste e negative. Per visualizzare il conteggio e la percentuale di una sezione, passa il mouse su di essa.



Crea una visualizzazione delle entità

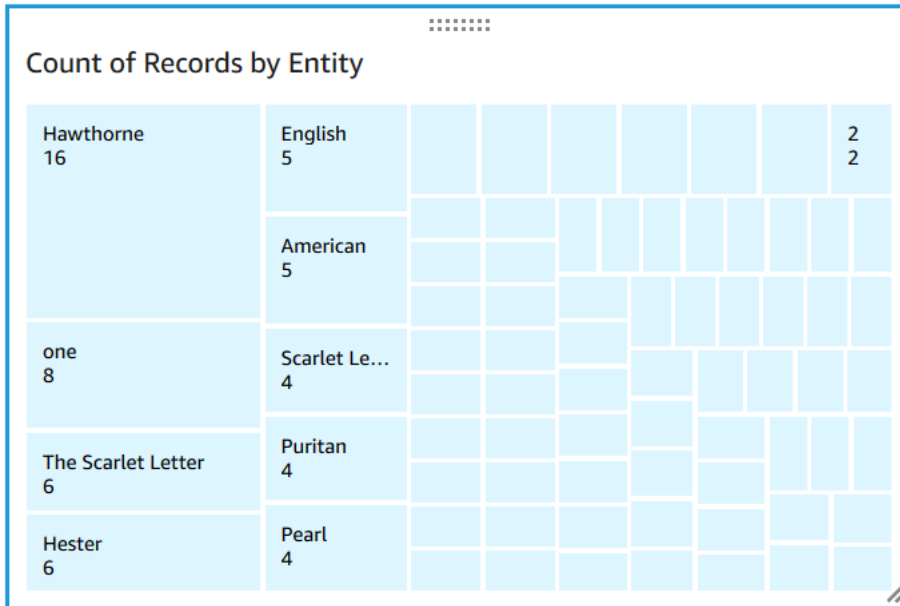
Ora crea una seconda visualizzazione con il set di dati delle entità. Si crea una mappa ad albero delle entità distinte nei dati. Ogni blocco nella mappa ad albero rappresenta un'entità e la dimensione del blocco è correlata al numero di volte in cui l'entità appare nel set di dati.

Per visualizzare i dati delle entità

1. Nel pannello di controllo Visualizza, accanto a Set di dati, scegli l'icona Aggiungi, modifica, sostituisci e rimuovi set di dati.
2. Scegli Aggiungi set di dati.
3. Per Scegli il set di dati da aggiungere, scegli il set di dati delle entità **entities_results_final** dall'elenco dei set di dati e scegli Seleziona.
4. Nel pannello di controllo Visualizza, scegli il menu a discesa Set di dati e scegli il set di dati delle entità. **entities_results_final**

5. Nell'elenco Campi, scegli Entità.
6. Per i tipi visivi, scegli Mappa ad albero.

Accanto al grafico a torta viene visualizzata una mappa ad albero simile alla seguente. Per vedere il conteggio di un'entità specifica, passa il mouse su un blocco.



Pubblica una dashboard

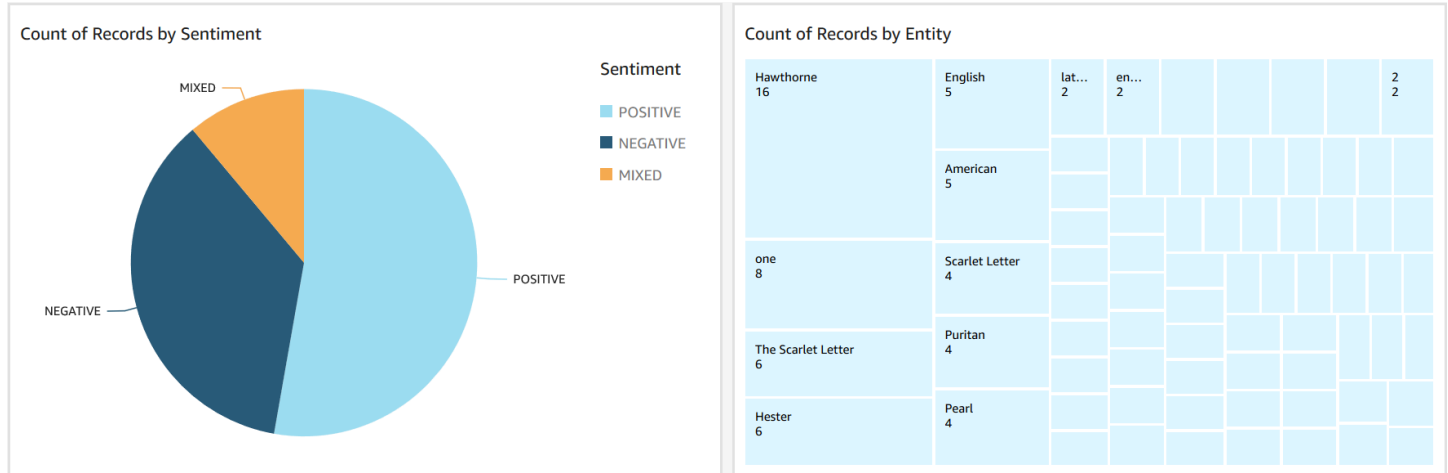
Dopo aver creato le visualizzazioni, puoi pubblicarle come dashboard. È possibile eseguire diverse attività con una dashboard, ad esempio condividerla con gli utenti che la utilizzano Account AWS, salvarla come PDF o inviarla tramite e-mail come report (limitata all'edizione Enterprise di Quick). In questa fase, pubblica le visualizzazioni come dashboard nel tuo account.

Per pubblicare la dashboard

1. Nella barra di navigazione, scegli Condividi.
2. Scegliere Publish dashboard (Pubblica pannello di controllo).
3. Scegli Pubblica nuova dashboard come e inserisci il nome comprehend-analysis-reviews per la dashboard.
4. Scegliere Publish dashboard (Pubblica pannello di controllo).
5. Chiudi il riquadro Condividi dashboard con gli utenti scegliendo il pulsante di chiusura nell'angolo in alto a destra.

- Nella Console rapida, nel riquadro di navigazione, scegli Dashboard. Una miniatura della nuova dashboard **comprehend-analysis-reviews** dovrebbe apparire in Dashboard. Scegli la dashboard per visualizzarla.

Ora hai una dashboard con visualizzazioni di sentimenti ed entità simili all'esempio seguente.



Tip

Se desideri modificare le visualizzazioni nella dashboard, torna a Analisi e modifica la visualizzazione che desideri aggiornare. Quindi, pubblica nuovamente la dashboard come nuova dashboard o come sostituto della dashboard esistente.

Eliminazione

Dopo aver completato questo tutorial, potresti voler ripulire tutte AWS le risorse che non desideri più utilizzare. AWS Le risorse attive possono continuare a comportare addebiti sul tuo account.

Le seguenti azioni possono aiutare a prevenire addebiti continui:

- Annulla l'abbonamento Quick. Quick è un servizio di abbonamento mensile. Per annullare l'abbonamento, consulta [Annullamento dell'abbonamento](#) nella Guida rapida per l'utente.
- Elimina il tuo bucket Amazon S3. Amazon S3 ti addebita lo storage. Per ripulire le tue risorse Amazon S3, elimina il bucket. Per informazioni sull'eliminazione di un bucket, vedi [Come si elimina un bucket S3?](#) nella Guida per l'utente di Amazon Simple Storage Service. Assicurati di salvare tutti i file importanti prima di eliminare il bucket.

- Cancella il tuo. AWS Glue Data Catalog Ti AWS Glue Data Catalog addebita mensilmente per lo spazio di archiviazione. Puoi eliminare i tuoi database per evitare di incorrere in addebiti continui. Per informazioni sulla gestione dei AWS Glue Data Catalog database, consulta [Lavorare con i database sulla AWS Glue console nella Guida](#) per gli AWS Glue sviluppatori. Assicurati di esportare i dati prima di cancellare qualsiasi database o tabella.

Utilizzo dei punti di accesso Lambda a oggetti Amazon S3 per informazioni di identificazione personale (PII)

Usa i punti di accesso Amazon S3 Object Lambda per le informazioni di identificazione personale (PII) per configurare il modo in cui i documenti vengono recuperati dal tuo bucket Amazon S3. Puoi controllare l'accesso ai documenti che contengono PII e cancellare le PII dai documenti. Per ulteriori informazioni su come Amazon Comprehend può rilevare le informazioni personali nei tuoi documenti, consulta. [Rilevamento delle entità PII](#) Gli access point Amazon S3 Object Lambda utilizzano AWS Lambda funzioni per trasformare automaticamente l'output di una richiesta Amazon S3 GET standard. Per ulteriori informazioni, consulta [Transforming objects with S3 object Lambda nella](#) Amazon Simple Storage Service User Guide.

Quando crei un punto di accesso Amazon S3 Object Lambda per informazioni personali, i documenti vengono elaborati utilizzando le funzioni di Amazon Comprehend Lambda per controllare l'accesso ai documenti che contengono informazioni personali e rimuovere le informazioni personali dai documenti.

Quando crei un punto di accesso Amazon S3 Object Lambda per informazioni personali, i documenti vengono elaborati utilizzando le seguenti funzioni di Amazon Comprehend Lambda:

- `ComprehendPiiAccessControlS3ObjectLambda`- Controlla l'accesso ai documenti con informazioni personali archiviate nel tuo bucket S3. Per ulteriori informazioni su questa funzione Lambda, accedi a per visualizzare la Console di gestione AWS ObjectLambda funzione [ComprehendPiiAccessControlS3](#) in. AWS Serverless Application Repository
- `ComprehendPiiRedactionS3ObjectLambda`- Elimina le informazioni personali dai documenti nel tuo bucket Amazon S3. Per ulteriori informazioni su questa funzione Lambda, accedi a per visualizzare la Console di gestione AWS ObjectLambda funzione [ComprehendPiiRedactionS3](#) in. AWS Serverless Application Repository

Per informazioni su come distribuire applicazioni serverless da AWS Serverless Application Repository, vedere [Deploying applications nella Serverless Application AWS](#) Repository Developer Guide.

Argomenti

- [Controllo dell'accesso ai documenti con informazioni di identificazione personale \(PII\)](#)
- [Oscurazione delle informazioni di identificazione personale \(PII\) dai documenti](#)

Controllo dell'accesso ai documenti con informazioni di identificazione personale (PII)

Puoi utilizzare un punto di accesso Amazon S3 Object Lambda per controllare l'accesso ai documenti con informazioni di identificazione personale (PII).

Per garantire che solo gli utenti autorizzati abbiano accesso ai documenti che contengono informazioni personali archiviate nel tuo bucket Amazon S3, utilizzi la funzione.

`ComprehendPiiAccessControlS3ObjectLambda` Questa funzione Lambda utilizza l'[ContainsPiiEntities](#) operazione durante l'elaborazione di una richiesta Amazon S3 GET standard su oggetti documento.

Ad esempio, se nel bucket S3 sono presenti documenti che includono informazioni personali come numeri di carte di credito o dati di conti bancari, puoi configurare la `ComprehendPiiAccessControlS3ObjectLambda` funzione per rilevare questi tipi di entità PII e limitare l'accesso agli utenti non autorizzati. Per ulteriori informazioni sui tipi di entità PII supportati, consulta. [Tipi di entità universali PII](#)

Per ulteriori informazioni su questa funzione Lambda, accedi a per visualizzare la Console di gestione AWS ObjectLambda funzione [ComprehendPiiAccessControlS3](#) in. AWS Serverless Application Repository

Creazione di un punto di accesso Lambda a oggetti Amazon S3 per controllare l'accesso ai documenti

L'esempio seguente crea un punto di accesso Amazon S3 Object Lambda per controllare l'accesso ai documenti che contengono numeri di previdenza sociale.

Creazione di un punto di accesso Lambda a oggetti Amazon S3 utilizzando AWS Command Line Interface

Crea una configurazione del punto di accesso Amazon S3 Object Lambda e salva la configurazione in un file chiamato `config.json`.

```
{
  "SupportingAccessPoint": "s3-default-access-point-name-arn",
  "TransformationConfigurations": [
    {
      "Actions": [
        "s3:GetObject"
      ],
      "ContentTransformation": {
        "AwsLambda": {
          "FunctionArn": "comprehend-pii-access-control-s3-object-lambda-arn",
          "FunctionPayload": "{\"pii_entities_types\": \"SSN\"}"
        }
      }
    }
  ]
}
```

L'esempio seguente crea un punto di accesso Amazon S3 Object Lambda in base alla configurazione definita nel file `config.json`

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

```
aws s3control create-banner-access-point \
  --region region \
  --account-id account-id \
  --name s3-object-lambda-access-point \
  --configuration file://config.json
```

Richiamo di un punto di accesso Lambda a oggetti Amazon S3 per controllare l'accesso ai documenti

L'esempio seguente richiama un punto di accesso Amazon S3 Object Lambda per controllare l'accesso ai documenti.

Richiamo di un punto di accesso Lambda a oggetti Amazon S3 utilizzando AWS Command Line Interface

L'esempio seguente richiama un punto di accesso Amazon S3 Object Lambda utilizzando il. AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```
aws s3api get-object \  
  --region region \  
  --bucket s3-object-lambda-access-point-name-arn \  
  --key object-prefix-key output-file-name
```

Oscurazione delle informazioni di identificazione personale (PII) dai documenti

Puoi utilizzare un punto di accesso Amazon S3 Object Lambda per oscurare le informazioni di identificazione personale (PII) dai documenti.

Per oscurare i tipi di entità PII dai documenti archiviati in un bucket S3, usi la funzione.

`ComprehendPiiRedactionS3ObjectLambda` Questa funzione Lambda utilizza [DetectPiiEntities](#) le operazioni [ContainsPiiEntities](#) and durante l'elaborazione di una richiesta Amazon S3 GET standard su oggetti documento.

Ad esempio, se i documenti nel tuo bucket S3 includono informazioni personali come numeri di carte di credito o dati di conti bancari, puoi configurare la `ComprehendPiiRedactionS3ObjectLambda` funzione per rilevare le PII e quindi restituire una copia di questi documenti in cui i tipi di entità PII sono oscurati. Per ulteriori informazioni sui tipi di entità PII supportati, consulta. [Tipi di entità universali PII](#)

Per ulteriori informazioni su questa funzione Lambda, accedi a per visualizzare la Console di gestione AWS ObjectLambda funzione [ComprehendPiiRedactionS3](#) in. AWS Serverless Application Repository

Creazione di un punto di accesso Lambda a oggetti Amazon S3 per eliminare le informazioni personali dai documenti

L'esempio seguente crea un punto di accesso Amazon S3 Object Lambda per redigere i numeri delle carte di credito dai documenti.

Creazione di un punto di accesso Lambda a oggetti Amazon S3 utilizzando AWS Command Line Interface

Crea una configurazione del punto di accesso Amazon S3 Object Lambda e salva la configurazione in un file chiamato `config.json`

```
{
  "SupportingAccessPoint": "s3-default-access-point-name-arn",
  "TransformationConfigurations": [
    {
      "Actions": [
        "s3:GetObject"
      ],
      "ContentTransformation": {
        "AwsLambda": {
          "FunctionArn": "comprehend-pii-redaction-s3-object-lambda-arn",
          "FunctionPayload": "{\"pii_entities_types\": \"CREDIT_DEBIT_NUMBER\"}"
        }
      }
    }
  ]
}
```

L'esempio seguente dimostra la creazione di un punto di accesso Amazon S3 Object Lambda basato sulla configurazione definita nel `config.json`

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (`\`) al termine di ogni riga con un accento circonflesso (`^`).

```
aws s3control create-access-point-for-object-lambda \
  --region region \
  --account-id account-id \
  --name s3-object-lambda-access-point \
  --configuration file://config.json
```

Richiamo di un punto di accesso Lambda a oggetti Amazon S3 per eliminare le informazioni personali dai documenti

Gli esempi seguenti richiamano un punto di accesso Amazon S3 Object Lambda per eliminare le informazioni personali dai documenti.

Richiamo di un punto di accesso Lambda a oggetti Amazon S3 utilizzando AWS Command Line Interface

L'esempio seguente richiama un punto di accesso Amazon S3 Object Lambda utilizzando il. AWS CLI

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituisci il carattere di continuazione UNIX barra rovesciata (\) al termine di ogni riga con un accento circonflesso (^).

```
aws s3api get-object \  
  --region region \  
  --bucket s3-object-lambda-access-point-name-arn \  
  --key object-prefix-key output-file-name
```

Soluzione: analisi del testo con Amazon Comprehend e OpenSearch

AWS fornisce un'implementazione di riferimento dell'analisi del testo utilizzando Amazon Comprehend e il OpenSearch servizio. Amazon Comprehend fornisce analisi del testo e OpenSearch fornisce indicizzazione, ricerca e visualizzazione dei documenti.

Per ulteriori informazioni, consulta [Analyzing text with OpenSearch and Amazon](#) Comprehend.

Guida di riferimento alle API

Il riferimento all'API è ora un documento separato. Per ulteriori informazioni, consulta [Amazon Comprehend API](#) Reference.

Cronologia dei documenti per Amazon Comprehend

La tabella seguente descrive la documentazione per questa versione di Amazon Comprehend.

Modifica	Descrizione	Data
Formazione personalizzata sui classificatori con documenti nativi	Amazon Comprehend ora supporta la formazione personalizzata sui classificatori con documenti nativi. Per ulteriori informazioni, consulta Modelli di classificazione della formazione in Amazon Comprehend .	19 aprile 2023
Flywheels per la gestione di modelli personalizzati	Amazon Comprehend ora supporta i volani per aiutarti a gestire l'addestramento e il monitoraggio delle versioni dei modelli per i modelli personalizzati. Per ulteriori informazioni, consulta Flywheels in Amazon Comprehend .	28 febbraio 2023
Argomenti di sicurezza IAM aggiornati	Sono stati aggiornati gli argomenti sulla sicurezza IAM per includere le identità federate. Per ulteriori informazioni, consulta Identity and Access Management for Amazon Comprehend .	22 dicembre 2022
Elaborazione in un'unica fase per l'inferenza con modelli personalizzati	Amazon Comprehend ora esegue automaticamente l'estrazione del testo per i documenti di input di immagini, PDF o Word prima di eseguire	1 dicembre 2022

la classificazione personali
zzata o il riconoscimento
personalizzato delle entità. Per
ulteriori informazioni, consulta
[Elaborazione dei documenti in
Amazon Comprehend](#).

[Sincrono APIs per un sentiment mirato](#)

Amazon Comprehend ora
supporta l'analisi sincrona
APIs e da console in tempo
reale per un sentiment mirato.
Il sentimento mirato determina
il sentimento associato a entità
specifiche in un documento
. Per ulteriori informazioni,
consulta [Targeted sentiment in
Amazon Comprehend](#).

21 settembre 2022

[Riduci il numero minimo di annotazioni per gli addetti alla formazione](#)

Amazon Comprehend ha
ridotto i requisiti minimi per la
formazione di un riconoscitore
con file di annotazione CSV
in testo semplice. Ora puoi
creare un modello di riconosci-
mento delle entità personali
zzato con un minimo di tre
documenti annotati e almeno
25 annotazioni per tipo di
entità. Per ulteriori informazi-
oni, vedere [Preparazione dei
dati di addestramento](#).

3 agosto 2022

Maggiore dimensione del documento di input in tempo reale APIs	Amazon Comprehend ora supporta un documento di input fino a 100 KB per la maggior parte dei documenti in tempo reale. APIs Per ulteriori informazioni, consulta Linee guida e quote .	18 luglio 2022
Tipi di entità PII aggiuntivi	Altri tipi di entità PII ora rilevati da Amazon Comprehend. Per ulteriori informazioni, consulta Rilevamento delle entità PII in Amazon Comprehend .	20 maggio 2022
Ristrutturazione del sommario	Ha ristrutturato il sommario della Amazon Comprehend Developer Guide per una navigazione più semplice. Per ulteriori informazioni, consulta What is Amazon Comprehend .	7 aprile 2022
Sentimento mirato	Amazon Comprehend ora supporta l'analisi mirata del sentiment, che determina il sentiment associato a entità specifiche in un documento . Per ulteriori informazioni, consulta Targeted sentiment in Amazon Comprehend .	9 marzo 2022
Nuova funzionalità	Amazon Comprehend ora ti consente di analizzare le immagini per il riconoscimento personalizzato delle entità. Per ulteriori informazioni, consulta Rilevamento di entità personalizzate in Amazon Comprehend .	28 febbraio 2022

Nuova funzionalità

Ora puoi copiare modelli personalizzati addestrati tra Account AWS. Per ulteriori informazioni, consulta [Copiare modelli personalizzati tra account in Amazon Comprehend](#).

2 febbraio 2022

Nuova funzionalità

Ora puoi utilizzarlo AWS Trusted Advisor per visualizzare consigli che possono aiutarti a ottimizzare i costi e la sicurezza degli endpoint Amazon Comprehend. Per ulteriori informazioni, consulta [Using Trusted Advisor with Amazon Comprehend](#).

29 settembre 2021

Nuova funzionalità

Amazon Comprehend ha lanciato una suite di funzionalità per Comprehend Custom che consentono miglioramenti continui del modello dandoti la possibilità di creare nuove versioni del modello, eseguire test continui su set di test specifici ed eseguire migrazioni in tempo reale verso nuovi endpoint del modello.

21 settembre 2021

Nuova funzionalità

Amazon Comprehend ora ti consente di analizzare documenti PDF e Word per il riconoscimento personalizzato delle entità. Con i formati PDF e Word, puoi estrarre informazioni da documenti contenenti intestazioni, elenchi e tabelle.

14 settembre 2021

Nuova funzionalità

Amazon Comprehend ha lanciato una nuova funzionalità di panoramica degli endpoint che offre una visione globale degli endpoint. Dalla pagina di panoramica degli endpoint, puoi visualizzare tutti gli endpoint in un unico posto per comprendere l'utilizzo degli endpoint rispetto all'utilizzo effettivo delle risorse.

24 agosto 2021

Nuova caratteristica

Amazon Comprehend Medical ora ti consente di stabilire una connessione privata con il tuo Virtual Private Cloud (VPC) creando un endpoint VPC di interfaccia. Per ulteriori informazioni, consulta [VPC endpoints](#) (). PrivateLink

13 giugno 2021

Espansione della lingua

Amazon Comprehend ha aggiunto quattro lingue aggiuntive per la funzionalità della lingua dominante: hausa (ha), lao (lo), maltese (mt) e oromo (om). Per ulteriori informazioni, consulta [Lingue supportate in Amazon Comprehend](#).

10 maggio 2021

Nuova funzionalità

Con Amazon Comprehend, ora puoi crittografare modelli personalizzati utilizzando una chiave gestita dal cliente (CMK). Per ulteriori informazioni, consulta la sezione [Crittografia KMS in Amazon Comprehend](#).

31 marzo 2021

Nuova funzionalità

Ora puoi utilizzare i punti di accesso Amazon S3 Object Lambda per configurare il modo in cui i documenti che contengono informazioni di identificazione personale (PII) vengono recuperati dal tuo bucket Amazon S3. Puoi controllare l'accesso ai documenti che contengono PII e cancellare le PII dai documenti. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Lambda a oggetti Amazon S3 per informazioni di identificazione personale](#) (PII).

18 marzo 2021

Nuova funzionalità

Ora puoi etichettare un documento con informazioni di identificazione personale (PII). Amazon Comprehend può analizzare il documento per verificare la presenza di informazioni personali e restituire le etichette dei tipi di entità PII identificati come nome, indirizzo, numero di conto bancario o numero di telefono. Per ulteriori informazioni, consulta [Etichettare un documento](#) con dati PII.

11 marzo 2021

Nuova funzionalità

Con Amazon Comprehend, ora puoi rilevare eventi in un set di documenti. Quando crei un processo di rilevamento di eventi asincrono, Amazon Comprehend è in grado di rilevare i tipi di eventi finanziari supportati. [Per ulteriori informazioni, consulta Rileva eventi.](#)

24 novembre 2020

Nuova funzionalità

Amazon Comprehend ora consente di utilizzare la scalabilità automatica per endpoint di riconoscimento di entità personalizzati. Con la scalabilità automatica, puoi impostare automaticamente il provisioning degli endpoint in base alle tue esigenze di capacità. Per ulteriori informazioni, consulta [Scalabilità automatica](#) con endpoint.

28 settembre 2020

Nuova funzionalità

Per addestrare classificatori personalizzati o riconosciitori di entità, ora puoi fornire file manifest aumentati, che sono set di dati etichettati prodotti da Amazon AI Ground Truth. SageMaker [Per ulteriori informazioni su questi file e per esempi, consulta Modalità multiclasse, Modalità multi-etichetta e Annotazioni.](#)

22 settembre 2020

Nuovo tutorial

Amazon Comprehend ora dispone di un tutorial che ti guida attraverso un flusso di lavoro multiservizio di analisi delle recensioni dei clienti e visualizzazione dei risultati dell'analisi. Per ulteriori informazioni, consulta [Tutorial: Analyzing insights from reviews.](#)

17 settembre 2020

Nuova funzionalità

Con Amazon Comprehend, ora puoi rilevare le entità nel testo che contengono informazioni di identificazione personale (PII), come indirizzi, numeri di conto corrente o numeri di telefono. Amazon Comprehend può fornire la posizione di ogni entità PII nel testo oppure può fornire una copia del testo in cui le PII sono redatte. Per ulteriori informazioni, consulta [Rilevare informazioni di identificazione personale](#) (PII).

17 settembre 2020

Nuova funzionalità

In precedenza, era possibile addestrare un modello solo su un massimo di 12 entità personalizzate. Ora Amazon Comprehend ti consente di addestrare un modello su un massimo di 25 entità personalizzate alla volta. Per ulteriori informazioni, consulta [Riconoscimento personalizzato delle entità](#).

12 agosto 2020

Espansione della lingua

Amazon Comprehend ha aggiunto cinque lingue aggiuntive per la funzionalità di riconoscimento personalizzato delle entità: tedesco (de), spagnolo (es), francese (fr), italiano (it) e portoghese (pt). Per ulteriori informazioni, consulta [Lingue supportate in Amazon Comprehend](#).

12 agosto 2020

Nuova funzionalità

Amazon Comprehend ora ti consente di stabilire una connessione privata con il tuo Virtual Private Cloud (VPC) creando un endpoint VPC di interfaccia. Per ulteriori informazioni, consulta [VPC endpoints](#) ().AWS PrivateLink

11 agosto 2020

Nuova funzionalità

Con Amazon Comprehend, ora puoi rilevare rapidamente entità personalizzate in singoli documenti di testo eseguendo analisi in tempo reale. Per ulteriori informazioni, consulta [Rilevamento di entità personalizzate in tempo reale con Amazon Comprehend](#).

9 luglio 2020

Nuova caratteristica aggiunta

Amazon Comprehend ora fornisce il supporto per una seconda modalità di classificazione personalizzata asincrona dei documenti che offre una maggiore flessibilità nell'applicazione di classi personalizzate ai documenti. Mentre la modalità multiclasse associa solo una singola classe a ciascun documento, la nuova modalità multi-etichetta può associarne più di una. Ad esempio, un film può essere classificato sia come fantascienza che come azione allo stesso tempo. Per ulteriori informazioni, consulta [Modalità multiclasse e multietichetta nella classificazione personalizzata](#).

19 dicembre 2019

Nuova caratteristica aggiunta

Amazon Comprehend ora fornisce supporto per la classificazione personalizzata in tempo reale per documenti con testo non strutturato. I clienti possono utilizzare la classificazione personalizzata in tempo reale per comprendere, etichettare e indirizzare le informazioni in base alle proprie regole aziendali, in modo sincrono. Per ulteriori informazioni, consulta [Analisi in tempo reale con classificazione personalizzata](#).

25 novembre 2019

Nuovi linguaggi aggiunti

Amazon Comprehend ha aggiunto sei lingue aggiuntive per molte delle sue funzionalità: arabo (ar), hindi (hi), giapponese (ja), coreano (ko), cinese semplificato (zh) e cinese tradizionale (zh-TW). Queste nuove lingue sono supportate solo per le operazioni Determine Sentiment, Detect Key Phrases e non personalizzate di Detect Entities. [Per ulteriori informazioni, consulta Lingue supportate](#).

6 novembre 2019

Nuova funzionalità

In precedenza, era possibile addestrare un modello solo su una singola entità personalizzata. Di conseguenza, era possibile cercare solo quell'entità con un'operazione di riconoscimento delle entità. Amazon Comprehend ha cambiato questa situazione e ora puoi addestrare un modello su un massimo di 12 entità personalizzate alla volta. Per ulteriori informazioni, consulta [Riconoscimento personalizzato delle entità](#)

09 luglio 2019

Nuova funzionalità

Amazon Comprehend ora fornisce una matrice di confusione multiclasse per una maggiore capacità di analisi dei parametri durante la formazione di un classificatore personalizzato. Questa funzionalità è attualmente supportata solo con. APIs Per ulteriori informazioni, consulta [Tagging resources in Amazon Comprehend](#)

5 aprile 2019

Nuova funzionalità

Amazon Comprehend fornisce tag per Custom Classifiers e Custom Entity Recognizer, che possono essere utilizzati come metadati per organizzare, filtrare e controllare l'accesso alle risorse con un livello di controllo più preciso che mai. Per ulteriori informazioni, consulta [Tagging resources in Amazon Comprehend](#)

3 aprile 2019

Nuova funzionalità

Amazon S3 ti consente già di crittografare i documenti di input e Amazon Comprehend estende ulteriormente questa funzionalità. Utilizzando la tua chiave KMS, puoi non solo crittografare i risultati di output del tuo lavoro, ma anche i dati sul volume di storage collegato all'istanza di calcolo che elabora il processo di analisi. Il risultato è la sicurezza. end-to-end Per ulteriori informazioni, consulta la sezione [Crittografia KMS in Amazon Comprehend](#)

28 marzo 2019

Nuova funzionalità

Il riconoscimento personali zzato delle entità estende le funzionalità di Amazon Comprehend consentendoti di identificare nuovi tipi di entità non supportati come uno dei tipi di entità generici preimpostati. Ciò significa che puoi analizzare documenti ed estrarre entità come codici di prodotto o entità specifiche dell'azienda che soddisfano le tue esigenze particolari. Per ulteriori informazioni, consulta Riconoscimento [personali zzato](#) delle entità

16 novembre 2018

Nuova funzionalità

Puoi usare Amazon Comprehend per creare modelli personalizzati per la classificazione personalizzata, assegnando un documento a una classe o a una categoria . Per ulteriori informazioni, consulta [Classificazione dei documenti](#).

15 novembre 2018

Espansione regionale

Amazon Comprehend è ora disponibile in Europa (Francoforte) (eu-central-1).

10 ottobre 2018

Espansione della lingua	Oltre all'inglese e allo spagnolo, Amazon Comprehend ora può esaminare documenti anche in francese, tedesco, italiano e portoghese. Per ulteriori informazioni, consulta Lingue supportate in Amazon Comprehend .	10 ottobre 2018
Espansione regionale	Amazon Comprehend è ora disponibile in Asia Pacifico (Sydney) (ap-southeast-2).	15 agosto 2018
Nuova funzionalità	Amazon Comprehend ora analizza i documenti per scoprire la sintassi di un documento e la parte del discorso per ogni parola. Per ulteriori informazioni, consulta Syntax .	17 luglio 2018
Nuova funzionalità	Amazon Comprehend ora supporta l'elaborazione in batch asincrona per il rilevamento di linguaggi o, frase chiave, entità e sentiment. Per ulteriori informazioni, consulta Elaborazione asincrona in batch .	27 giugno 2018
Nuova guida	Questa è la prima versione della Amazon Comprehend Developer Guide.	29 novembre 2017

AWS Glossario

Per la AWS terminologia più recente, consultate il [AWS glossario](#) nella sezione Reference. Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.