



Università degli Studi di Padova
Ingegneria del Software
Anno Accademico: 2021/2022



Bug's Bunny

Verbale esterno

2022/05/16

Redazione: Marco Bellò
Verifica: Giulio Zanatta
Approvazione: Angela Arena



Generalità

- Ora inizio: 14.00
- Ora fine: 16.00

Presenze

- Angela Arena
- Marco Bellò
- Tommaso Di Fant
- Ruth Genevieve Bousapnamene
- Matteo Tossuto
- Marco Volpato
- Giulio Zanatta

Resoconto

Le predizioni che andremo a fare non avranno bisogno di essere precise al 100% (ovviamente sarebbe impossibile).

Consiglio (dell'esperto) per apprendere gli strumenti di AWS: cercare di apprendere iterativamente, non tutto subito, perchè sono strumenti molto complessi.

Rekognition

Non necessita di esperienza nell'ambito Machine Learning per poterlo usare, è tutto già pronto (training e quant'altro) nel caso in cui si usino modelli già pronti. Se si vuole costruire un modello custom è richiesta tanta più esperienza quanto si vuole "deviare" dallo standard.

Ha la caratteristica di scalare automaticamente (come lambda praticamente) e di integrarsi con tutti gli altri servizi di AWS.

Fornisce molti tipi di analisi: oggetti e scene, comparazione e riconoscimento volti, ricerca volti in una collezione, riconoscimento celebrità, moderazione di immagini, ecc...

È possibile fare binding per vari linguaggi, la demo sarà in Python (usando boto3).

Labels

I labels vengono forniti come risultato testo facilmente comprensibile, e hanno una struttura di massima simile a questa:

```
1 'Name' : 'Vehicle',
2 'Confidence' : 0.something,
3 'Instances' : [...]
```



Instances è vuoto se il label è riguardante l'intera immagine, altrimenti contiene le coordinate del/dei bounding box.

Non tutte le labels sono indipendenti, alcune sono correlate fra loro, per esempio Vehicle fa parte della classe Transportation.

Bounding Box

Servono a dare una localizzazione spaziale in un'immagine quando sono presenti più istanze di uno stesso label. Definisce una sottoarea dell'immagine determinata secondo delle coordinate: il vertice superiore sinistro dell'immagine è definito come "0", mentre il vertice inferiore destro come "1". Questo permette di creare un sistema di coordinate dove $[x,y]$ sono sempre compresi tra 0 e 1.

A questo punto si usano 4 valori per definire il BB:

- Left: 0.qualcosa (distanza dal lato sinistro)
- Top: 0.qualcosa (distanza dal lato superiore)
- Height: 0.qualcosa (altezza BB)
- Width: 0.qualcosa (larghezza BB)

Inoltre è presente anche il dato della confidence, che si riferisce ovviamente alla label che il BB rappresenta.

Label detection

Ho vari tipi di labels, come ad esempio:

- Beach (scene)
- Outdoors (concept)
- Person (object)
- Running (action)

e posso trovarli sia su immagini intere, che su frazioni dell'immagine (chiamati bounding box). Ho circa 2580 labels già esistenti per immagini intere e 239 per i bounding box.

Custom Labels

Se non bastano le labels fornite, se ne possono aggiungere altre, riaddestrando il modello, dovendo però fornire il dataset. Il modo più semplice per fare dei custom labels è tramite transfer learning: prendo un modello già allenato e lo adatto mano mano alle mie esigenze. È buona prassi fare un PoC del modello dopo un incremento del dataset (seguito da fase di training) per verificare se serva lavorarci ancora o se vada bene.

Codice S3 + Bytes → Arn è il codice univoco fornito da Amazon per identificare un nuovo modello (che idealmente ho fatto io).

Quante immagini servono? Almeno 50 per ogni label, e devono essere quanto più diversificate possibile.

S3 + Rekognition standard



```
1 client = boto3.client('rekognition')
2 response = client.detect_labels\{
3     Image = {'S3 Object' : {
4         'Bucket' : bucket,
5         'Name' : photo}
6     },
7     MaxLables = 10
8 }
```

Posso settare un upper limit del numero di labels che voglio (MaxLabels), inoltre posso settare anche una MinConfidence per scremare risultati dei quali il modello non è molto sicuro.



Bytes + Rekognition

```
1 import boto3
2 client = boto3.client('rekognition')
3 response = client.detect_labels(Image = {'Bytes': image.read()})
```

Console

Region → cerco di prendere un server che sia quanto più vicino possibile al mio S3.

Searchbar → Rekognition → use Custom Labels

Mi viene quindi richiesto di fare un bucket in S3 dove salvare i modelli.

Poi:

1. Create Project
2. Create Dataset (import images from S3 bucket)

Inoltre avere i labels con lo stesso nome dei folders aiuta in quanto è il comportamento visto come "di default" da AWS.

Vengono fatti degli esempi di label detection, prendendo le immagini da s3 oppure direttamente da un file su file system. Le operazioni di AWS Rekognition possono analizzare immagini salvate in S3 Bucket oppure passate tramite bytes (e quindi salvate localmente).

Best Practices

Best practice: se l'immagine è già presente in S3, logicamente sarà più veloce di Bytes. In caso contrario, nel quale non sia presente in S3, non è conveniente fare l'upload di quest'ultima e poi l'analisi, conviene utilizzare Bytes. La Region Zone va scelta opportunamente in base alla locazione dei servizi che si andranno ad utilizzare per ottimizzare le prestazioni degli algoritmi. Infine sarebbe meglio spegnere il modello quando non è in uso.

Comprehend

Lavora sui testi, in particolare permette di eseguire task di sentiment analysis tramite NLP. Come rekognition, gestisce automaticamente le risorse e non necessita di conoscenze nell'area. Ricerca all'interno di testi di: entità (nomi, oggetti, luoghi,...), key phrases (info rilevanti nel contesto del documento), dati identificativi detti PII (indirizzi, numeri di telefono,...), lingua dominante, sentiment (positivo, negativo, neutro o misto), sentiment per ogni entity, sintassi.

Nota: non tutte queste cose sono disponibili in tutte le lingue.

Come per rekognition, permette di definire classificazioni e entity custom.

Lingue

Targeted sentiment è disponibile solo in inglese, mentre Sentiment c'è anche in italiano. Non tutti i servizi sono presenti in tutte le lingue.



Document clustering

Organizza documenti basandosi su caratteristiche comuni e similitudini, unsupervised learning.

Modalità di processamento

- Sincrono singolo documento
- Sincrono multi documento
- Asincrono in batch, con molti documenti

Quando viene fatta l'analisi, viene ritornata una risposta contenete la probabilità (sentiment score) per mixed, neutral, positive e negative. La somma delle probabilità è (ovviamente) 1.

```
1 import boto3
2 comprehend = boto3.client(service_name='comprehend', region_name='
    region')
3 comprehend.detect_sentiment(Text = text, LanguageCode = 'en')
```

Output:

```
1 "Sentiment score": {
2     "Mixed" : [0,1]
3     "Negative" : [0,1]
4 }
5 "Sentiment" : "NEUTRAL"
```

Dove Sentiment è il valore più alto tra tutti, ogni Sentiment ha valore tra 0 e 1, i quali sommati danno 1.

Console

Comprehend → Analysis Job → Create Job

Access Permissions: prestare attenzione a questo punto perchè ricco di cose da impostare (ovviamente presente config di default).