



TEDx

4all

Compito 2: AWS Glue & PySpark

Pesenti Luca (mat. 1079602)

Pesenti Alessandro (mat. 1082457)



Tecnologie Cloud e Mobile 2024





INDICE

1 Watch next

2 Transcript

3 Criticità



JOB PYSPARK - Watch next

```
# CREATE THE AGGREGATE MODEL
images_dataset = images_dataset.groupBy(col("id").alias("id_ref")).agg(collect_list(col("url").alias("url_images")))
#images_dataset = images_dataset.select(col("id").alias("id_ref"), \
#                                     col("url").alias("url_image"))
```

```
# READ RELATED_VIDEOS DATASET
relatedvideo_dataset_path = "s3://tedx-data/related_videos.csv"
relatedvideo_dataset = spark.read.option("header", "true").csv(relatedvideo_dataset_path)

relatedvideo_dataset = relatedvideo_dataset.drop("id")

rel_video_info = tedx_dataset_main.select(col("url"), \
                                         col("description"), \
                                         col("publishedAt"), \
                                         col("interalId"), \
                                         col("url_images"))
```

```
# ADD ALL INFORMATION OF EACH RELATED VIDEO
relatedvideo_dataset = relatedvideo_dataset.join(rel_video_info, relatedvideo_dataset.related_id == rel_video_info.interalId, "left")

# CREATE THE AGGREGATE MODEL
relatedvideo_dt = relatedvideo_dataset.groupBy(col("interalId").alias("id_from")).agg(collect_list(struct(col("title"), \
                                                         col("presenterDisplayName").alias("speaker"), \
                                                         col("duration"), \
                                                         col("url"), \
                                                         col("description"), \
                                                         col("publishedAt"), \
                                                         col("url_images"))).alias("related_videos")))

# AND JOIN WITH THE MAIN TABLE
tedx_dataset_agg = tedx_dataset_agg.join(relatedvideo_dt, tedx_dataset_agg.interalId == relatedvideo_dt.id_from, "left") \
    .select(col("id").alias("_id"), col("**")) \
    .drop("id_from") \
    .drop("id")
```

- Alcuni video presentano immagini in diversi formati. Le abbiamo anch'esse raggruppate in un array per evitare problematiche durante le operazioni successive.
- Recuperiamo le informazioni dei vari video ed andiamo a selezionare solo quelle andranno mostrate nei video collegati (watch next). Effettuiamo questi ulteriori passaggi in modo da fornire uno schema dettagliato dei video collegati, includendo dati utili come l'URL.
- Dopo aver recuperato l'id di ogni video collegato aggiungiamo le relative informazioni. Ogni video avrà quindi una lista di video watch next suggeriti da TED.

COLLECTION - Watch next

Esempio di elemento nella collection tedx_data

```
_id: "153616"
slug: "petter_johansson_do_you_really_know_why_you_do_what_you_do"
speakers: "Petter Johansson"
title: " Do you really know why you do what you do?"
url: "https://www.ted.com/talks/petter_johansson_do_you_really_know_why_you_"
description: "Experimental psychologist Petter Johansson researches choice blindness..."
duration: "960"
publishedAt: "2018-03-06T16:00:08Z"
interalId: "10361"
url_images: Array (3)
  0: "https://talkstar-photos.s3.amazonaws.com/uploads/b9fa704c-e3a3-4af3-84..."
  1: "https://talkstar-photos.s3.amazonaws.com/uploads/4954b34c-d0cf-4d04-a8..."
  2: "https://talkstar-photos.s3.amazonaws.com/uploads/68a1069d-193a-4c17-a3..."
tags: Array (7)
  0: "cognitive science"
  1: "psychology"
  2: "illusion"
  3: "decision-making"
  4: "self"
  5: "magic"
  6: "TEDx"
related_videos: Array (6)
  0: Object
    title: "The paradox of choice"
    speaker: "Barry Schwartz"
    duration: "1163"
    url: "https://www.ted.com/talks/barry_schwartz_the_paradox_of_choice"
    description: "Psychologist Barry Schwartz takes aim at a central tenet of western so..."
    publishedAt: "2006-09-26T00:11:00Z"
    url_images: Array (2)
  1: Object
  2: Object
  3: Object
  4: Object
  5: Object
```

Dopo l'esecuzione dello script precedente, un documento della collection `tedx_data` ha la seguente struttura:

- un array contenente le immagini di copertina nei vari formati
- un array contenente i tag
- un array di video collegati e le relative informazioni utili

JOB PYSPARK - Transcript

```
# READ TRANSCRIPT_VIDEOS DATASET
transcript_dataset_path = "s3://tedx-data/transcripts.csv"
transcript_dataset = spark.read \
    .option("header","true") \
    .option("quote", "\"") \
    .option("escape", "\\") \
    .option("multiline","true") \
    .csv(transcript_dataset_path)
```

```
# CREATE THE AGGREGATE MODEL
transcript_dataset_agg = transcript_dataset.groupBy(col("id").alias("id_ref_transcript"))\
    .agg(collect_list(struct(col("time"), col("text"))).alias("transcript"))
```

```
# AND JOIN WITH THE MAIN TABLE
tedx_dataset_agg = tedx_dataset_agg.join(transcript_dataset_agg, tedx_dataset_agg.id == transcript_dataset_agg.id_ref_transcript, "left") \
    .select(col("id").alias("_id"), col("*")) \
    .drop("id_ref_transcript") \
    .drop("id")
```

Scraper - TEDx Transcript Scraper

- Importa il file details.json trasformato precedentemente dal csv fornitoci
- Accede tramite referenza (interallId) all'API GraphQL fornita da TED restituendo tramite query i timestamp ed il testo associato.
- Se esiste ricava la trascrizione, formata da una o più frasi. Ogni frase ha associato l'istante in cui lo speaker inizia a pronunciarla.

Script - Load Transcript

- Aggiunge ai talk la loro trascrizione
- Vengono indicati come liste di oggetti dalla struttura (time,text)

COLLECTION - Transcript

Esempio di elemento nella collection tedx_data

```
_id: "112048"
slug: "ben_wellington_how_we_found_the_worst_place_to_park_in_new_york_city_u..."
speakers: "Ben Wellington"
title: "How we found the worst place to park in New York City -- using big dat..."
url: "https://www.ted.com/talks/ben_wellington_how_we_found_the_worst_place_..."
description: "City agencies have access to a wealth of data and statistics reflectin..."
duration: "698"
publishedAt: "2015-02-26T15:58:03Z"
interalId: "2199"
url_images: Array (2)
tags: Array (7)
transcript: Array (272)
  0: Object
    time: "711"
    text: "Six thousand miles of road,"
  1: Object
    time: "3531"
    text: "600 miles of subway track,"
  2: Object
    time: "5734"
    text: "400 miles of bike lanes"
  3: Object
    time: "7378"
    text: "and a half a mile of tram track,"
  4: Object
    time: "9199"
    text: "if you've ever been to Roosevelt Island."
  5: Object
  6: Object
  7: Object
  8: Object
  9: Object
  10: Object
```

Dopo l'esecuzione dello script precedente, aggiungiamo una lista contenente i vari timestamp (in millisecondi) delle frasi pronunciate dallo speaker. Risulteranno poi utili in futuro per rendere le traduzioni dei sottotitoli più efficaci e puntuali.

CRITICITÀ



Debugging Non è stato possibile trovare un metodo semplice ed efficace per il debugging del codice.

I log disponibili si sono rivelati poco utili e confusi.

Inoltre, l'attesa di diversi minuti per verificare l'assenza di errori e analizzare l'output ha rallentato significativamente il processo di sviluppo di un codice ottimale ed efficiente.


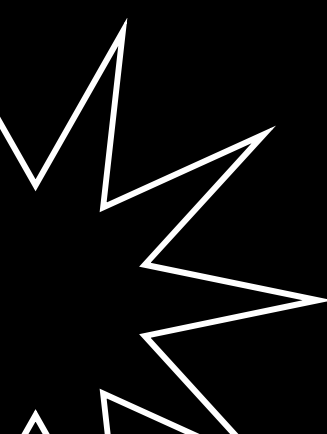


Complessità I vari codici identificativi e le relative relazioni referenziali sono risultate confusionarie e sarebbero potute essere più chiare tra i vari file csv, facilitando quindi le varie operazioni di giunzione.



POSSIBILI EVOLUZIONI

Sarà necessario collegare la nostra base di dati con lo strumento AWS Amazon Translate in modo da poter rendere fruibili i contenuti anche a coloro che non conoscono la lingua inglese. Come anticipato precedentemente, essendo la trascrizione divisa in timestamp, potremo effettuare una ricostruzione dei sottotitoli efficiente e rapida.





TED4all

Ponte tra culture e lingue



[Trello Board](#)



[GitHub Repository](#)

