Ingegneria Dei Dati - Homework 2

Davide Molitierno - 537969

21 ottobre 2022

Link github del progetto: https://github.com/chila99/IDDHomework2

1 Introduzione

Il progetto in questione ha lo scopo di indicizzare dei documenti in formato .txt relativi ai testi di canzoni di differenti artisti e di fornire la possibilità per mezzo di apposite query di effettuare ricerche verso l'indice creato. Nello specifico il sistema è costituito da due differenti classi, **Indexing** e **Searching**, che verranno analizzate nelle sezioni successive e che hanno rispettivamente il compito di implementare le due funzioni sopracitate.

2 Dataset

A seguito di numerose difficoltà nel trovare un dataset corposo di file testuali, la cartella dei documenti è stata generata attraverso l'utilizzo del seguente script Python.

```
import lyricsgenius

genius = lyricsgenius.Genius(personal_token, verbose=True,
    remove_section_headers=True, skip_non_songs=True, retries
    =5, timeout=120)
artists = ["Imagine Dragons", "Ed Sheeran" "Justin Bieber", "
    AC/DC", "Bad Bunny", "Beyonce", "Tiziano Ferro"]
for artist in artists:
    retrievedArtist = genius.search_artist(artist, max_songs
    =100, get_full_info=False)
```

```
    for song in retrievedArtist.songs:
        with open('../songs/'+ slugify(song.title) + '.txt',
'w') as f:
            f.write(retrievedArtist.name +"\n")
            f.write(song.lyrics)
```

Tale script si avvale della libreria Genius di lyricsGenius e permette di ottenere e salvare su una directory in locale 100 lyrics di una lista di artisti passata come parametro(la funzione slugify effettua il parsing della stringa passata in input, in questo caso il titolo della singola canzone, andando ad eliminare caratteri che successivamente avrebbero potuto creare problemi).

3 Indexing

Tale classe ha lo scopo, a partire dal path della directory in cui sono salvati i diversi file, di indicizzare i documenti sulla base di due differenti campi:

- titolo: contenente il titolo della canzone.
- contenuto: contenente il testo della canzone.

L'intera directory contiene 573 file testuali di dimensioni non eccessivamente elevate ed il sistema impiega circa 300ms per effettuare l'intera operazione di indicizzazione. Infine la classe di indexing si avvale di due tipi differenti di analyzer:

- un CustomAnalyzer con tokenizer di tipo whitespace e un filtro di tipo capitalization per il campo titolo in modo tale che tutti i token di tale campo vengano preservati e la loro prima lettera resa maiuscola;
- uno StandardAnalyzer con una serie di stopwords della lingua inglese, italiana e spagnola per il campo contenuto così da rendere meno dipendente la ricerca dei documenti da tali token di uso comune.

4 Searching

Tale classe permette in loop(fino a che non si inserisce il valore 0) di effettuare query di diverso tipo ottenendo differenti risultati. Nello specifico verranno ritornati sempre i primi 10 risultati nel ranking per il quale viene fatto matching. Le query permesse seguono la seguente sintassi: "query" e

“campo: query”. La prima permette di effettuare la ricerca su tutti i campi mentre la seconda esclusivamente sul campo che viene specificato. Qualora non si inserisca alcuna stringa il sistema non restituisce alcun documento. Infine è anche possibile fornire al programma una phrasequery inserendo tra apici(“”) la frase da voler cercare. Nota: quando vengono effettuate ricerche per il campo titolo è stato usato un analyzer speciale che rende maiuscole tutte le iniziali presenti nella query.

La tabella 1 mostra esempi di query con risultati:

Queries		
Query	**Descrizione**	**Primi 3 Risultati**
love	query su tutti i campi	Love / Love Hungry / Man Hold Up
titolo: love	query sul titolo	Love / Love Song / Love Drough
contenuto: love	query sul contenuto	Love / Love Hungry Man / Hold Up
love hate	or query su tutti i campi	Broken-Hearted Girl / COZY / Enemy(SoloMix)
hate OR blood	specific or query su tutti i campi	Believer / If You Want Blood... / Ring the Alarm
hate AND blood	AND query su tutti i campi	Believer
titolo: blood contenuto: hate	query specifica su entrambi i campi	If You Want Blood Youve Got It
titolo: ”love on top”	phrase query sul titolo	Love on Top
contenuto: ”ridi ridi ridi”	phrase query sul contenuto	Ma So Proteggerti
“”	stringa vuota	nessun documento

Table 1: Esempi di query.