Social Media Management

YOUTUBE

Autore
Giorgio Umberto
Gambino

Matricola
X81000851

UNIVERSITÀ
degli STUDI
di CATANIA

Contents

1 Introduzione

1.1 Obiettivo

L'obiettivo è quello di analizzare alcuni video su youtube ed estrarre alcune informazioni che potrebbero servire al proprietario del canale. In particolare cercheremo di ottenere due tipi di informazioni:

1. Per ogni utente cercheremo di ottenere la percentuale di positività che ogni utente ha su quel canale.
2. Estrarre le parole più frequenti per ogni utente.

2 Strumenti Utilizzati

2.1 Linguaggi e framework

Per il backend del progetto è stato utilizzato Python, dove faremo uso di alcune librerie. Es. nltk, usata nelle applicazioni di natural language processing (NLP) e data scientist. Mentre per il frontend è stato utilizzato Flask, iniziato come un semplice wrapper intorno a Werkzeug e Jinja ed è diventato uno dei framework di applicazioni web Python più popolari.

2.2 Macchina e OS

Il processore che monta su questa macchina è un Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz, possiede 16,0 GB(15,8 GB usable) di RAM e ha una GPU NVIDIA GEFORCE 1050 4GB GDDR5. Il sistema operativo è Windows 10 Home Version 21H2.

3 Esperimento

3.1 Backend

I files che si occupano del backend sono main.py e youtube.py.

3.2 Frontend

I files del frontend invece, si trovano nella cartella templates.

3.3 Librerie utilizzate

Le librerie utilizzate sono le seguenti:

```
import re
import pandas as pd
import nltk
import tqdm
import os
import numpy as np
import openpyxl
import json
from pyyoutube import Api
from numpy import dtype, float16, float32, int64
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.feature_extraction.text import TfidfVectorizer
from flask import Flask, redirect, url_for, request,
                                  render_template, session
```

Pyyoutube è un wrapper che usa le API di Youtube, pandas e openpyxl le utilizzerò per la gestione dei dati e per la lettura/scrittura sui file excel, flask verrà usata per realizzare il frontend, nltk e sklearn verranno usate per la realizzazione dei task.

3.4 Prima task

L'intento è quello di mostrare attraverso una dashboard, la percentuale di positività per ogni singolo utente. In più per arricchire un pò la tabella, aggiungerò altre informazioni, come il nome dell'autore, il numero dei commenti positivi, negativi e neutrali, il conteggio dei commenti e il conteggio dei mi piace.
Quindi, realizzerò mediante la seguente funzione un file xlsx, inizialmente vuoto, che conterrà le informazioni sopra elencante.

```
# Create an empty xlsx file which contain some information,
# AUTHOR, POSITIVE, NEUTRAL, NEGATIVE, COUNTCOMMENTS, COUNTLIKES
def create_file_sentence_user(self):

    if os.path.isfile(self.path_sentence_user):
        print ("File" + self.sentence_user + " exist")
    else:
        df = pd.DataFrame(columns = ['AUTHOR', 'POSITIVE', 'NEUTRAL', 'NEGATIVE',
        'COUNTCOMMENTS','COUNTLIKES', 'POS %'])

        df = df.astype({'NEGATIVE':np.float32,'POSITIVE':np.float32,
                        'NEUTRAL':np.float32,'COUNTCOMMENTS':np.int,
                        'COUNTLIKES':np.int, 'POS %':np.float16})
        df.to_excel(self.path_sentence_user, index = False)
        print ("File" + self.sentence_user + " created")
```

Per ogni video, estraggo delle informazioni e attraverso vader, calcolerò il sentiment su ciascun commento in modo da ottenere s la compound è una metrica che calcola la somma di tutte le valutazioni lessicali che sono state normalizzate tra -1 (negativo più estremo) e +1 (positivo più estremo).
Sentiment positivo : (punteggio composto >= 0,05) sentiment neutrale : (punteggio composto > -0,05) e (punteggio composto < 0,05) e sentiment negativo: (punteggio composto <= -0.05).
La Sentiment Analysis è il processo di determinazione "computazionale" se un pezzo di scrittura è positivo, negativo o neutro. È anche noto come opinion mining, derivando l'opinione o l'atteggiamento di un oratore.

```
# Export some information from the video,
# NEG, NEU, POS, COMPOUND are calculated by vader
def export_info_video_from_youtube(self, list_videos):
    for video in list_videos:
        comment = self.get_comment(video).to_dict()["items"]
        sid = SentimentIntensityAnalyzer()
        stats_comm = []
        for c in comment:
            id_top_comment = c["snippet"]["topLevelComment"]["id"]
            text_comment = c["snippet"]["topLevelComment"]["snippet"]["textOriginal"]
            like_comment = c["snippet"]["topLevelComment"]["snippet"]["likeCount"]
            num_comment = c["snippet"]["totalReplyCount"]
            name_user_comment = c["snippet"]["topLevelComment"]["snippet"]["authorDisplayName"]
            id_user_comment = c["snippet"]["topLevelComment"]["snippet"]["authorChannelId"]['value']
            sentiment_comment = sid.polarity_scores(text_comment)
            sentence_comment = self.get_sentence(sentiment_comment['compound'])

            stats_comm.append([video, id_top_comment, text_comment,
                               like_comment, num_comment, name_user_comment,
                               id_user_comment, sentiment_comment['neg'],
                               sentiment_comment['neu'], sentiment_comment['pos'],
                               sentiment_comment['compound'], sentence_comment])

        # Create an xlsx file with the extracted information
            df = pd.DataFrame(data = stats_comm, columns=["ID_VIDEO", "ID_TOP_COMMENT", "TEXT",
                                                  "AUTHOR","ID_AUTHOR", "NEG",
                                                  "NEU", "POS",
                                                  "COMPOUND" ,"SENTENCE"])
        df.to_excel(video + '.xlsx')
```

Una volta scaricati i video, ritorno un dataframe con le informazioni che mi servono.

```
# Return a df with some extracted information from the file xlsx,
# AUTHOR, TEXT, POS, NEG , NEU, COMOOUND, SENTENCE, LIKE
def export_info_video_from_excel(self, video):
    directory_path = os.getcwd()
    files_name = directory_path + "\\videos\\" + video
    df = pd.read_excel(files_name, sheet_name="Sheet1", usecols="D,E,G,H:K,M",
    dtype={'ID_AUTHOR':str,'NEG':float32,'POS':float32,'NEU':float32,'SENTENCE':str} )

    return df[["AUTHOR", "TEXT", "POS", "NEG", "NEU", "SENTENCE", "LIKE"]]
```

Adesso, per ogni singolo utente, mi calcolo il numero di commenti positivi, negativi e neutrali che ha lasciato, questo è possibile attraverso il compound. Riempio il file Sentence.xlsx, senza calcolarne la positività.

```
def calculate_all_sentence_for_each_user(self, df, my_sentence_user):
    # For each author count the sentences and insert it into new file.
    for a in df['AUTHOR'].unique():

        comm_pos, comm_neg, comm_neu,
        countcomm, countlikes = self.count_sentence_user(df, a)

        # search if the author is present or not in the file
        index_author = self.find_author(my_sentence_user, a)

        # if not present, append
        if index_author == 0:
            my_sentence_user.append([a,comm_pos, comm_neu, comm_neg,
                                     countcomm, countlikes])
        # if present, update the values
        elif(index_author != 0):
            self.update_value_of_author(my_sentence_user, a, index_author,
                                        comm_pos, comm_neu, comm_neg,
                                        countcomm, countlikes)
```

```
 # Return the count of all comments,
 # the number of likes and the count of all positive, negative and neutral comments
def count_sentence_user(self, df, author):

    comm_pos, comm_neg, comm_neu, countcomm, countlikes = 0, 0, 0, 0, 0

    new_df = df[df['AUTHOR'] == author]

    for val, cnt in new_df['SENTENCE'].value_counts().iteritems():
        if val == 'positive':
            comm_pos = cnt
        elif val == 'negative':
            comm_neg = cnt
        else:
            comm_neu = cnt

        countcomm += cnt

    countlikes =  new_df.loc[:, 'LIKE'].sum()
    return comm_pos, comm_neg, comm_neu, countcomm, countlikes
```

Infine, avremo modo di calcolare la positività, in che modo? Fare un rapporto tra il numero di commenti positivi e il numero di commenti totali.

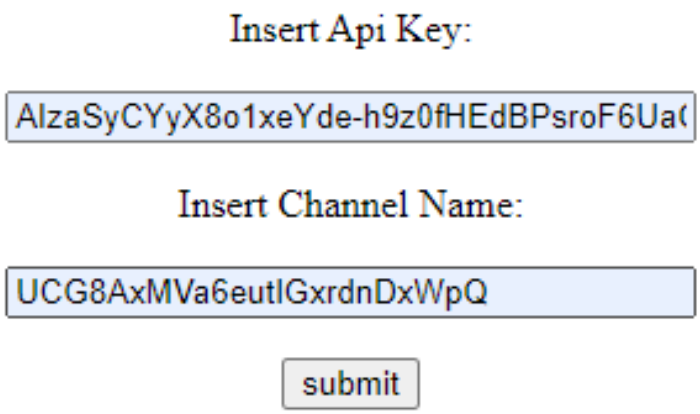
```
def calculate_positivity(self):
    sentence_user = openpyxl.load_workbook(self.path_sentence_user)
    my_sentence_user = sentence_user.active
    for a in range(1, len(my_sentence_user['A'])):
        my_sentence_user.cell(row = a + 1, column = 7).value =
        round((my_sentence_user.cell(row = a + 1, column = 2).value /
        my_sentence_user.cell(row = a + 1, column = 5).value) * 100)

    sentence_user.save(self.path_sentence_user)
    sentence_user.close()
```

Il tutto è racchiuso in una funzione situata nel main.

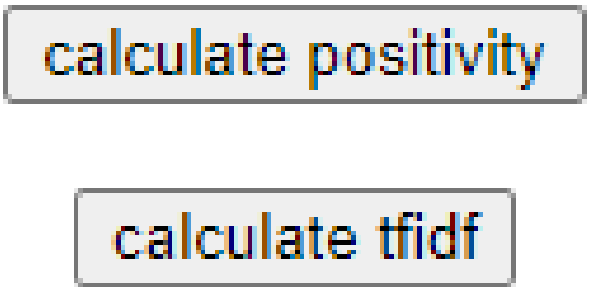
```
@app.route('/sentence/')
def sentence():
    obj = yt.Youtube(session['name_api'], session['name_channel'])
    videos = []
    # Get all videos in the folder
    for root, dirs, files in os.walk(os.getcwd() + "\\videos"):
        for filename in files:
            videos.append(filename)
    info_videos = {}

    obj.create_file_sentence_user()

    # Create and open the file Sentence User.xlsx
    sentence_user = openpyxl.load_workbook(obj.get_path_sentence_user())
    my_sentence_user = sentence_user.active
    for video in tqdm(videos):
        print("Export info from video", video)
        info_videos = obj.export_info_video_from_excel(video)
        print("Calculate all sentence for each user.")
        obj.calculate_all_sentence_for_each_user(info_videos, my_sentence_user)
```

3.5 Risultato della prima task

Eseguire il file main.py, e aprire il file login.html situato nella cartella templates. Aperto il file, inserisci la Api Key ed il nome del canale.



I video sono stati già scaricati, basta fare un drag and drop dei video che si vogliono analizzare dentro la cartella videos, dopo scegliamo l'operazione calculate positivity.



Risultato della dashboard ottenuta:

Sentence

Show 10 entries Search:

AUTHOR	POSITIVE	NEUTRAL	NEGATIVE	COUNTCOMMENTS	COUNTLIKES	POS %
#1RandomShorts	3	0	0	3	3	100
#kaya gaming	0	6	0	6	0	0
--	0	0	3	3	0	0
0000 0000	0	0	6	6	0	0
032 JAI VENKATESHH R	3	0	0	3	0	100
10Fr0sT01	0	0	6	6	0	0
10HP	0	6	0	6	6	0
16-bit Potato	0	3	0	3	0	0
2006jakebob	6	0	0	6	0	100
3A Animations	0	6	0	6	0	0

Showing 1 to 10 of 1,891 entries Previous 1 2 3 4 5 ... 190 Next

Potrebbe richiedere un pò di tempo per elaborare i video che sono presenti nella cartella.

3.6 Seconda Task

Per la realizzazione della seconda task, per prima cosa cercherò di estrarre per ogni utente, tutti i commenti.

```
def get_comments_of_all_users(self, id_video):

    path = os.getcwd() + "\\videos\\" + id_video + str(".xlsx")
    info_video = pd.read_excel(path, sheet_name='Sheet1')

    authors = info_video.ID_AUTHOR
    dictionary = {}

    # For each author return a list which contain all comments
    for a in authors.unique():
        comments = []
        for index, item in enumerate(authors):
            if(item == a):
                comments.append(str(info_video.iloc[index,3]))

        comments = [item.replace('\n',' ') for item in comments]
        #comments = [item.split(" ") for item in comments]

    # Merge all user comments to create a corpus
        result = ""
        for item in comments:
            result += item + " "


        dictionary[a] = result


    return dictionary
```

Successivamente andiamo a calcolare tfidf per ogni utente, ogni utente ha un proprio corpus, contenente tutti i commenti che ha rilasciato.

```
def create_tfidf_file_video(self, dictionary, id_video):
    path = os.getcwd() + "\\tfidf_video"
    if os.path.exists(path):
        # Apply tfidf
        response, features_names, index_dictionary = self.tfidf(dictionary)

       # Create a file excel with the values of tfidf
        df = pd.DataFrame(response.todense(), index = index_dictionary , columns = features_names)
        df.to_excel(path + "\\"+ "tfidf_" + id_video + ".xlsx")
    else:
        print("directory tfidf_video doesn't exist")
```

La tabella calcolata da tfidf, avrà come indici, i nomi degli utenti e come colonne avremo le words estratte.

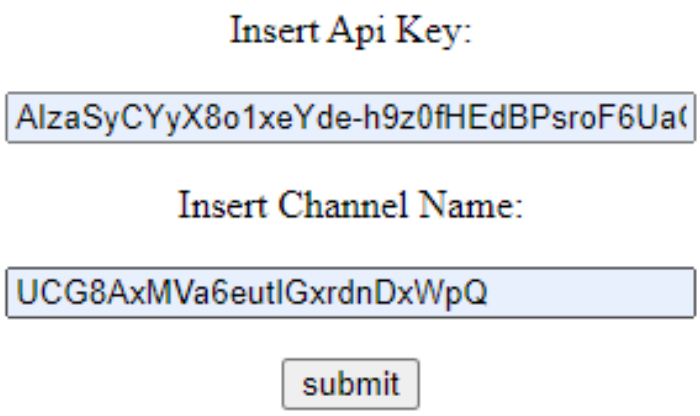
```
def tfidf(self, dictionary):
    # The regex does not consider emojis
    vectorize = TfidfVectorizer(token_pattern=r'([a-zA-Z0-9-/]{1,})'
                                ,analyzer='word', stop_words='english')
    response = vectorize.fit_transform(dictionary.values())
    features_names = vectorize.get_feature_names_out()
    index_dictionary = dictionary.keys()
    return response, features_names, index_dictionary
```

Infine estraiamo le parole, in che modo? Prendiamo tutti i valori delle parole dell'utente, e ne calcoliamo una media, infine estraiamo tutte le words che sono maggiori o uguali della media calcolata.

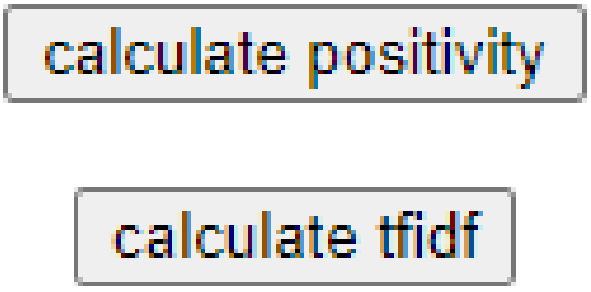
```
def extract_words_from_file_excel(self, id_video):

    tfidf = pd.read_excel(os.getcwd() + "\\tfidf_video\\" + "tfidf_" + id_video + ".xlsx",
                          dtype=object)
    dict_auth = []
    authors = tfidf.iloc[1:, 0]

    for name, values in authors.iteritems():
        list_couple_auth = []
        for idx, data in tfidf.iloc[name,1:].iteritems():
            if data != 0:
                list_couple_auth.append((idx, data))

        # Calculate avg
        avg = 0 if len(list_couple_auth) == 0 else
        (sum(ele[1] for ele in list_couple_auth) /len(list_couple_auth))

        # Extract words
        list_words_auth = []
        for ele in list_couple_auth:
            if ele[1] >= avg:
                list_words_auth.append(ele[0])

        dict_auth.append({"AUTHOR":values,"WORDS":list_words_auth})

    json_str = json.dumps(dict_auth, indent = 2)
    return json_str
```

3.7 Risultato seconda task

Eseguire il file main.py, e aprire il file login.html situato nella cartella templates. Aperto il file, inserisci la Api Key ed il nome del canale.



I video sono stati già scaricati e sono presenti nella cartella videos, basta fare il drag e drop di un singolo video, successivamente scegliamo l'operazione calculate tfidf.



Risultato finale, estratte le words frequenti per ogni utente.

Words

Show 10 entries Search:

AUTHOR	WORDS
Ranjeetkumar Yadav	youtubing
Teemu Leppä	yes
Arlyan yt	yea
Realguy809	yaay
ckjones1	wtf
BDFD'Mİ ?	wow
FurryMcFlurry	worth
ISA Ahmed	worst
a	wonder
Adriano Gaming	woah

Showing 1 to 10 of 712 entries

Previous 1 2 3 4 5 ... 72 Next

4 Conclusioni

Il progetto non è ottimizzato, potrebbe richiedere un pò di tempo per elaborare i dati. Nonostante ciò, viene comunque illustrato di come sia semplice lavorare con i dati e di come possano essere utili certi programmi per la gestione e la monitorazione dei propri utenti.