Natural Language Processing: Find The Most Common Phrases on Twitter

Tazki Anida Asrul · Follow

4 min read · Mar 1, 2019

GIF from here

Last Sunday, one of the biggest awards ceremonies was held. Yup, Oscar 2019. People all over the world got excited about this event. On Twitter, we could see a ton of tweets were posted before, during, and after the ceremony.

So… what the most discussed things during the Oscar this year? I tried to figure it out by using my basic knowledge about NLP and Python.

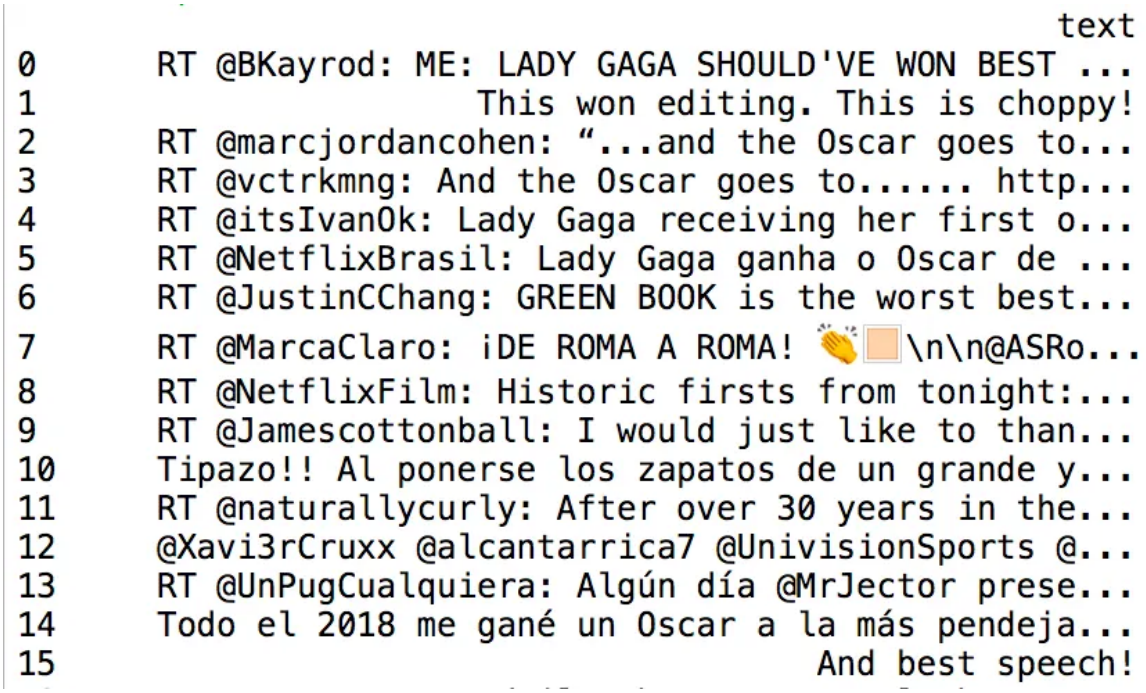
First of all, I scraped Twitter data using Tweepy. Tweepy library will help you to use Twitter Streaming API, so you can capture Twitter messages in real-time. In my case, I streamed the data for around 10 minutes, with 'oscar' as the filtered keyword. The results would be formed as a JSON file, and only tweets with the 'oscar' word on it would be extracted.

The second step was the most interesting and challenging part of all, **preprocessing**. In this step, I converted the unstructured data in JSON format to a more readable structure format, Pandas DataFrame.

```
import pandas as pd
import numpy as np
import json
import re
from nltk.corpus import stopwords
from nltk.util import ngrams
from collections import Counter

# For visualization:
import matplotlib.pyplot as plt
import seaborn as sns

tweets = []

for line in open('oscar.json', 'r'):
    try:
        content = json.loads(line)
        content.pop('limit', None)
        tweets.append(content)
    except:
        continue
```

```
data = pd.DataFrame(tweets, columns = ['text'])
data = data.dropna()
```

As seen below, the data still contain a lot of stop words, usernames, and even 'HTTP' links. We need to clean it out.



	text
0	RT @BKayrod: ME: LADY GAGA SHOULD'VE WON BEST ...
1	This won editing. This is choppy!
2	RT @marcjordancohen: "...and the Oscar goes to...
3	RT @vctrkmng: And the Oscar goes to..... http...
4	RT @itsIvanOk: Lady Gaga receiving her first o...
5	RT @NetflixBrasil: Lady Gaga ganha o Oscar de ...
6	RT @JustinCChang: GREEN BOOK is the worst best...
7	RT @MarcaClaro: ¡DE ROMA A ROMA! 👋🏻\n\n@ASRo...
8	RT @NetflixFilm: Historic firsts from tonight:...
9	RT @Jamescottonball: I would just like to than...
10	Tipazo!! Al ponerse los zapatos de un grande y...
11	RT @naturallycurly: After over 30 years in the...
12	@Xavi3rCruxx @alcantarrica7 @UnivisionSports @...
13	RT @UnPugCualquiera: Algún día @MrJector prese...
14	Todo el 2018 me gané un Oscar a la más pendeja...
15	And best speech!

So all I did was do some cleansing processes, including removing some patterns, punctuations, and English stop words.

```
#remove user, https, and RT
data['clean_tweet'] = np.vectorize(remove_pattern)(data['text'],
"https|RT|@[\w]*")

#remove punctuations
data['clean_tweet'] = data['clean_tweet'].str.replace("[^a-zA-Z#]", "
")

#lowering string
data['clean_tweet'] = data['clean_tweet'].str.lower()

#remove stop words
stop_words = set(stopwords.words('english'))
```

```
data['clean_tweet'] = [' '.join([w for w in x.lower().split() if w
not in stop_words])
    for x in data['clean_tweet'].tolist()]

#remove words with len < 2
data['clean_tweet'] = data['clean_tweet'].apply(lambda x: ' '.join([w
for w in x.split() if len(w)>2]))

#tokenization
tokenized_tweet = data['clean_tweet'].apply(lambda x:
list(ngrams(x.split(), 2)))
```

And the result would be like:

```
0       [(lady, gaga), (gaga, best), (best, actress), ...
1                                    [(editing, choppy)]
2       [(oscar, goes), (goes, green), (green, book), ...
3                     [(oscar, goes), (goes, dyedaxkp)]
4       [(lady, gaga), (gaga, receiving), (receiving, ...
```


If you notice, there are only the main words for each tweet. I captured the phrases of the main words by using the **Ngrams** function. Ngrams can help us to tokenize the list of the sentence by breaking up the string and grouping them into phrases. The second word in a tuple will be the first word in the following one, and so on. For example, if we have a sentence like *"lady gaga watch oscar"* as the input, the Ngrams output will be like:

```
[(lady, gaga), (gaga, watch), (watch, oscar)]
```

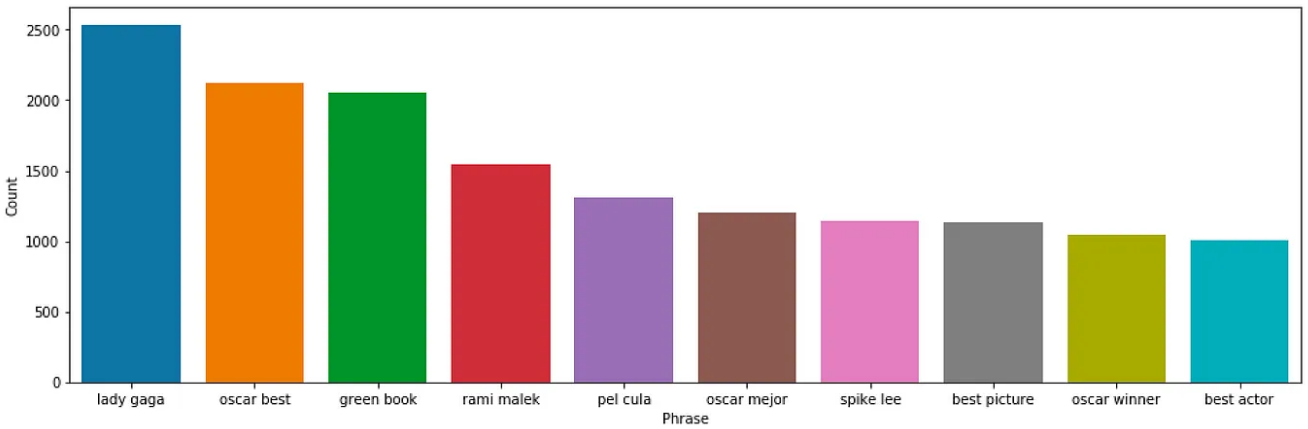
After getting the phrases from all tweets, I tried to combine them all into one list and count the appearance of each phrase.

```
l = reduce(lambda x, y: list(x)+list(y), zip(tokenized_tweet))
flatten = [item for sublist in l for item in sublist]

counts = Counter(flatten).most_common()

df = pd.DataFrame.from_records(counts, columns=['Phrase', 'Count'])
df['Phrase']= df['Phrase'].apply(lambda x: ' '.join([w for w in x]))
```

In the final step, I used Matplotlib and Seaborn library to visualize the result more beautifully.

```
df = df.nlargest(columns="Count", n = 10)
plt.figure(figsize=(15,4))
ax = sns.barplot(data=df, x= "Phrase", y = "Count")
ax.set(ylabel = 'Count')
plt.show()
```

I created a bar chart to present the 10 most common phrases, and the result is...



'Lady Gaga' got first place, followed by 'Oscar Best' and 'Green Book'. Overall, the most discussed topics were related to the actress/actor (Lady Gaga, Rami Malek), movie (Green Book), director (Spike Lee), and nomination (Best

Picture, Best Actor). We can also see some phrases are in Spanish because we didn't filter the tweets by language.

However, the experiment above is just a little example of text processing using Twitter data. There's still a lot of room to improve and explore the idea of NLP utilization. Hope it will help!

Reference

1. https://methodi.ca/recipes/analyzing-repeating-phrases-ngrams-python
2. https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/

Data Science | Text Mining | Data Processing

Written by Tazki Anida Asrul

228 Followers

Playing around with data

Follow

More from Tazki Anida Asrul

Tazki Anida Asrul in Towards Data Science

Turn Photos into Cartoons Using Python

You can give a cartoon effect to a photo by implementing machine learning algorithms i...

4 min read · Jan 2, 2021

1.8K 23

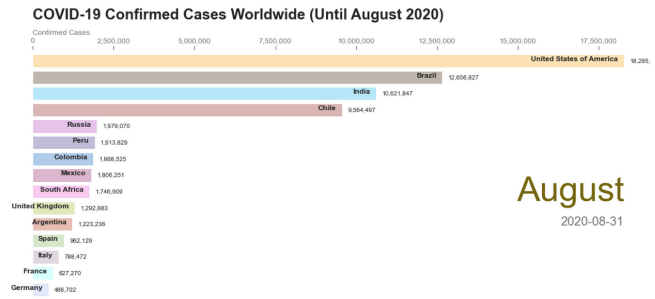
Tazki Anida Asrul

BigQuery and Apache Airflow: The Fundamental Tools for Data...

In any kind of business environments, data has played a major role to drive important...

5 min read · Jul 6, 2020

75



Tazki Anida Asrul in Towards Data Science

Analyzing Music Video Trends on Youtube Using Python

Find out the most popular music videos based on Youtube search trends.

6 min read · Nov 15, 2020

107 1

Tazki Anida Asrul in Towards Data Science

Python Bar Chart Race Animation: COVID-19 Cases

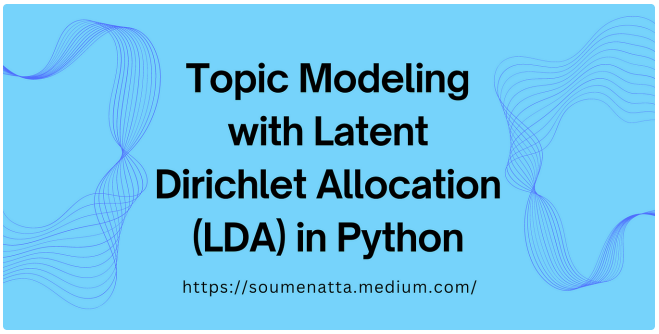
COVID-19 has crushed many countries for over eight months. Million cases has been...

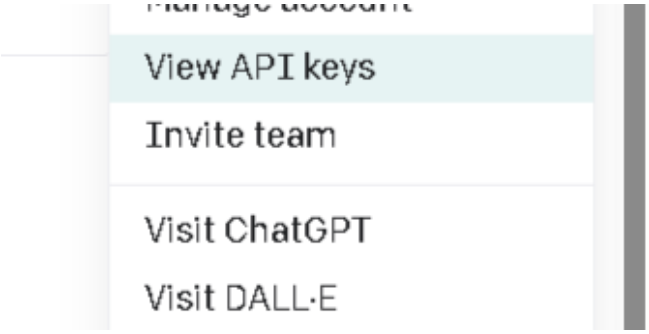
5 min read · Sep 18, 2020

187 2

See all from Tazki Anida Asrul

Recommended from Medium





Dr. Soumen Atta, Ph.D. in Level Up Coding

Topic Modeling with Latent Dirichlet Allocation (LDA) in Python

Topic Modeling is a natural language processing (NLP) technique used to discove...

✦ · 4 min read · Jul 25

62

Afzal Muhammad

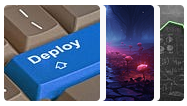
Extract data from news articles using ChatGPT in Python

Introduction

5 min read · Jul 7

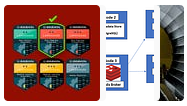
11

Lists



Predictive Modeling w/ Python

20 stories · 652 saves



New_Reading_List

174 stories · 209 saves

Practical Guides to Machine Learning

10 stories · 733 saves

Coding & Development

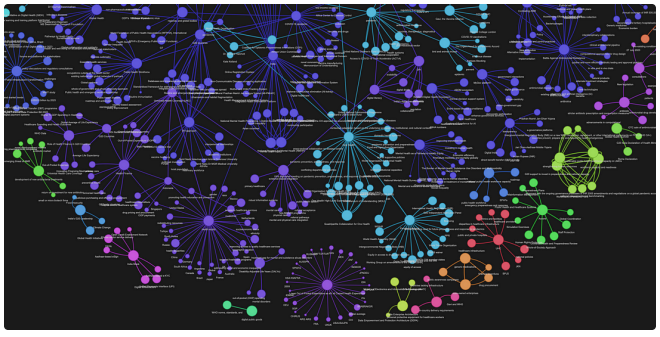
11 stories · 295 saves

Viola Mao

News Scraping with Python's Newspaper3k—A Step-by-Step...

Library newspaper3k is a Python web scraping framework mainly used for grabbin...

✦ · 6 min read · Sep 26



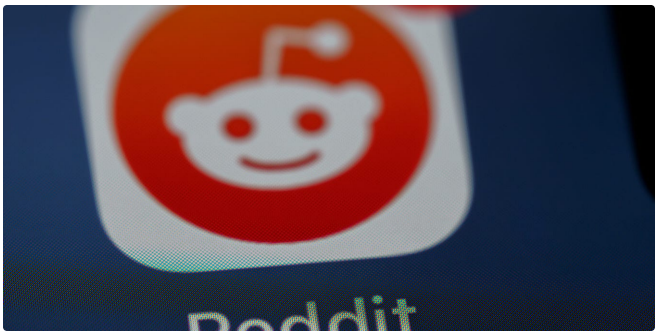
Rahul Nayak in Towards Data Science

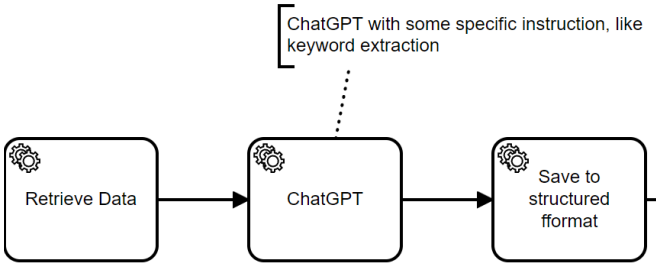
How to Convert Any Text Into a Graph of Concepts

A method to convert any text corpus into a Knowledge Graph using Mistral 7B.

12 min read · Nov 10

59 3.4K 39





Aziz Budiman in Data And Beyond

Sentiment Analysis and Topic Modelling of Reddit Headlines...

Analysing the sentiments and topic model on hot ChatGPT headlines using Natural...

· 6 min read · Jun 7

70

g gil fernandes

Keyword Extraction with LangChain and ChatGPT

In this blog we will try to explain how we can extract keywords using LangChain and...

4 min read · Jun 7

68

See more recommendations