twitstat

Release v0.0.1

Aditya Raman

2020-10-13

TABLE OF CONTENTS

1 About Twitstat **3**

2 Scraping Module **5**

3 Analysis Module **7**
- 3.1 Preprocessing Module 7
- 3.2 Clustering Module 7
- 3.3 Sentiment Analysis Module 7

4 Future Iterations **9**

5 Resources and References **11**

6 Contributors **13**

7 Indices and tables **15**

Contents:

CHAPTER

ONE

ABOUT TWITSTAT

Twitstat is a simple web application that analyses twitter data to provide interesting insights into trending hashtags and topics. It cleverly clusters and charts data to ease the process of better understanding trends around the world!

Twitstat is split into multiple modules

- *Scraping Module*
- *Analysis Module*

CHAPTER

TWO

SCRAPING MODULE

Twitstat uses Twitter's python API tweepy[1] to get all the tweets for the analysis. Tweepy is first used to fetch the trending topics around a specified geographical location, these fetched topics are then fed into the api's search method. The search method gets Twitstat all the tweets (and other important information such as the likes, retweets, et cetera for each tweet) corresponding to the search query.

[1] https://github.com/tweepy/tweepy

CHAPTER

THREE

ANALYSIS MODULE

Twitsat uses three major modules to facilitate its data analysis

- Preprocessing module
- Clustering module
- Sentiment analysis module

3.1 Preprocessing Module

Before data can be loaded into any of the *actual* analyser functions, it has to be preprocessed or *cleaned*. The preprocessing module removes any unwanted text such as emoticons, line breaks, punctuations et cetera, from the tweets. Certain words *(called stop-words)* are also removed as they do not add meaning to the text. At last, all the words are tokenized *(split into multiple words)* and *stemmed*. These tasks are done with the help of nltk's[2] algorithms.

3.2 Clustering Module

Twitstat's clustering module uses scikit-learn's[3] DBSCAN[4] clustering algorithm to cluster tweets falling under the trending categories. **Density-based spatial clustering of applications with noise** *(DBSCAN)* is a density-based clustering algorithm, that is, given a set of points in some space, it groups together points that are closely packed together. Points which are sparsely packed are classified as outliers.

3.3 Sentiment Analysis Module

At last, after splitting tweets into clusters, the most popular tweet of each cluster is identified. These *popular* tweets are then fed into texblob's[5] sentiment analysis module where the tone (positive, negative or neutral) of the tweets is decided.

[2] https://github.com/nltk/nltk
[3] https://github.com/scikit-learn/scikit-learn
[4] https://scikit-learn.org/stable/modules/clustering.html#dbscan
[5] https://github.com/sloria/TextBlob

CHAPTER

FOUR

FUTURE ITERATIONS

Twitter + Statistics = Amazing information!

And that is why, we want to keep improving. Future iterations of Twitstat will include (but are not limited to)

- A new and improved clustering algorithm to cluster data with higher fidelity
- Get better insights on data by geo-locating tweets and forming heat-maps
- Create gists of each modelled topic for a quick look into what's the most talked about in real time

CHAPTER

FIVE

RESOURCES AND REFERENCES

Twitstat and this documentation would have not been possible without these amazing resources!

- Scikit-learn clustering documentation[6]
- Tweepy documentation[7]
- This insightful paper![8]
- 'Text Mining and Clustering of Tweets Based on Context' by Toly Novik[9]
- Tutorial on Scikit-learn Tfi-df with nltk preprocessing[10]

And all the amazing open source software![11]

[6] https://scikit-learn.org/stable/modules/clustering.html
[7] http://docs.tweepy.org/en/latest/
[8] https://github.com/heerme/twitter-topics/blob/master/insight-snow14dc-final.pdf
[9] https://www.dezyre.com/student-project/toly-novik-text-mining-and-clustering-of-tweets-based-on-context/2
[10] https://www.bogotobogo.com/python/NLTK/tf_idf_with_scikit-learn_NLTK.php
[11] https://github.com/MLH-Fellowship/twitstat/blob/main/requirements/base.txt

CHAPTER

SIX

CONTRIBUTORS

Made with love by Aditya Raman[12] and Garima Singh[13]!

[12] https://github.com/ramanaditya
[13] https://github.com/grimmmyshini

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search