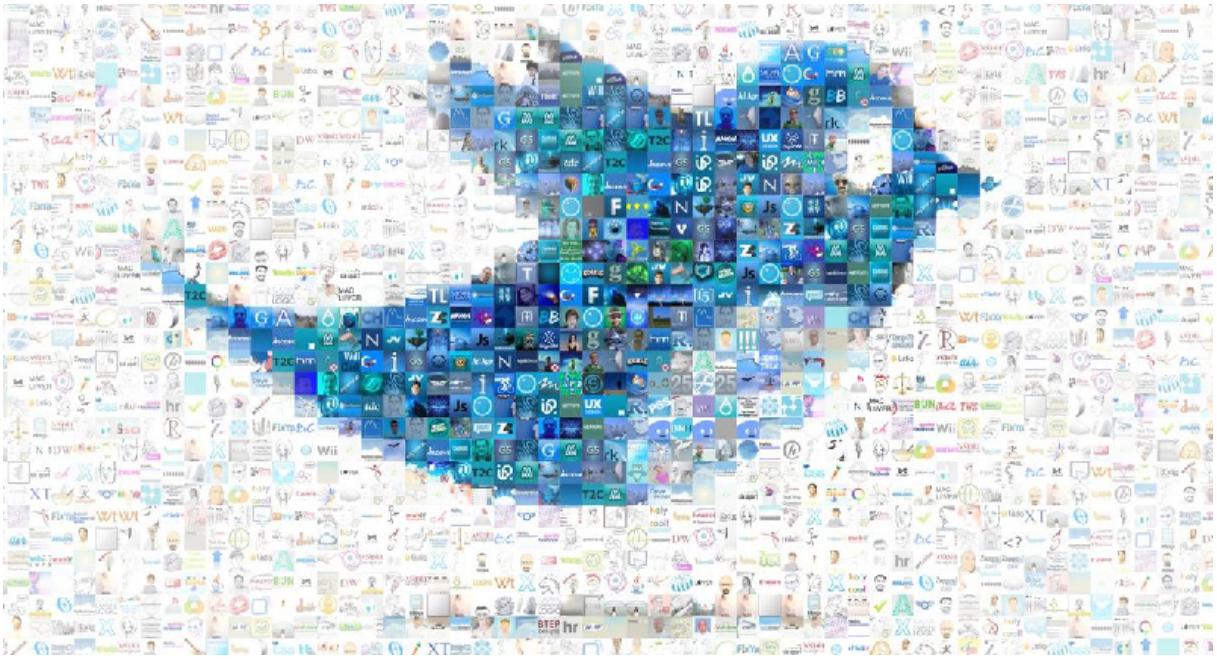
Report on Twitter topic detection and analysis

Alex Tsilingiris
Christina Pardalidou
Sotiris Karapostolakis

July 2, 2017



Contents

1 Introduction

This technical report as detailed below describes the implementation of a project regarding twitter topic detection and analysis. The main goal was (a) the detection of emerging events from a collected dataset crawled from twitter live streaming, (b) the extraction of sentiments concerning the top emerging event and (c) specifying the users' geolocation that were tweeting about this event. These were achieved using methodologies and techniques that are found in the literature. Hence, chapter 2 presents the implementation details for every topic, chapter 3 describes the framework in a more technical manner and chapter 4 depicts the results.

2 Method implementation details

Event detection

Our approach

A combination of approaches was used for event detection: Since the dataset was static and filtered based on a query (in our case the word ”trump”), terms were considered as living organisms in the given timespan, allowing us to sample a set of tweets as when each term was ”most alive”, while combining user reputation metrics to select a tweet from a reputable source.

Hashtags as terms

We solely operated on hashtags on this step since they represent an idea or a topic, which in other cases would be difficult, or not as accurate, to define using Natural Language Processing and Machine Learning.

The living organism implementation

We considered a term as most-alive at the point in which it was mostly detected in our dataset. The time window was set to 2 hours. Therefore, we had to query for the mostly found terms (using the Term Frequency statistic) and sort the results in descending order. We then generated the final that contained the original tweets with the most active terms in the dataset that were posted in the given timespan.

Selecting a tweet from the pool

We now have a pool of tweets that might contain information about an event. The approach we followed was to assume that a user with a high amount of followers represents an influential event source into a social community[9]. We sorted (descending) the tweet pool by the number of followers the original poster has and ended up with 1 tweet that was our result for each term.

Sentiment Analysis

Methodology

For the prediction of the sentiments of the tweets gathered, we used a supervised method. In particular, we used a machine-learning technique. The tool we used for the predictions was implemented for the needs of a course in the previous semester. Since we had put a lot of effort in its implementation and it is a creation of our personal work we decided to use it for this assignment too.

Selecting tweets from the dataset

All the tweets related to the top emerging event were isolated and written to different text files. These are all the tweets that include hashtags related to the event (e.g #parisagreement). All the retweets were removed.

Sentiment Prediction

The Opinion-Mining tool[3] we used implements a SVM classifier that classifies the tweets into two categories: positive and negative. SVM performs very well and is considered as one of the top performing classifiers in the literature. This tool is further presented in chapter 3.

Location Prediction

Geoinference

Twitter provides the users the ability to infer the location either by defining the place the user is or by enabling the location, where the position of the user is more accurate. The generic approach started by gathering, through a query to our database, both users' categories. We excluded the users with minor place inference, due to the lack of information that could lead to a correct prediction. The main idea was to form a model to foresee some of those users' location.

Dataset Creation

We created 4 different datasets so we can have the ability to compare the results. Initially we collected users' tweets having as a criterion a specific number (i.e 20) of the most inferenced locations. We noticed that all of them were in the USA. The second criterion was the number of users for each locality. Based on the second, we formulated the 4 different datasets.

Implementation

The methodology is based on a simple technique. For every user we calculated the frequency of every location's reference in his/her tweets. The highest number was given as a prediction. We consulted that a user is most probable to cite his/her city more frequent than any other place. The results were quite encouraging.

3 Framework description

The software created was split into modules to allow independent, on-demand functionality. A user can easily assemble a system from these independent modules based on their needs. The complete repository can be found in the project's Github page [2].

Getting the tweets

To begin with, Python and Tweepy [7] were used to access the Twitter API as efficiently as possible. The incoming stream was filter using the keyword "trump" and was directly saved into a MongoDb instance.

Preprocessing the dataset

A custom preprocessing module was created to combine all the required steps needed when analyzing tweets. The nltk [4] library along with some custom regular expressions were used in order to complete the following actions: group hashtag symbols with the hashtag word, group mention symbols with the mention target, keep URLs functional and finally remove stop-words, punctuation and twitter specifics such as "via" and "rt". Finally, a script was created to convert datetime strings into MongoDb specific datetime strings so that range queries were possible.

Plotting

The pandas [5] library was used to resample the time and allow tweet bucketing based on a given time window. The window size used for term (hashtag) visualization was 120minutes. The results were exported directly into .json format and are presented in our website [1] using the d3js [6] library.

Getting the tweets for sentiment analysis

After detecting the top event that occurred during the time the tweets were crawled, the dataset was filtered in order to get only the tweets, without the retweets, containing hashtags related to the emerging event (e.g #parisagreement, #climatechange). These specific tweets were written to text files so they can be imported to the Opinion-Mining tool[3] for sentiment prediction.

Geoprediction datasets

We mentioned the use of 4 different datasets. Now, we are going to explain the way of forming the datasets. The first step was to calculate the most referenced places. From those we selected the top-20 cities for the first, third and fourth dataset. We excluded places such as states or countries. The second step was the selection of a specific number of unique users for each dataset. The numbers were 20, 15, 25, 30 for

each corresponding dataset. Every time we picked the 200 latest tweets for each user. The algorithm was applied in every dataset.

Opinion-Mining tool

The tool uses a custom lexicon implementation that is shown in figure 3.1. The training set used includes a total of 25k labeled reviews taken from the published dataset of IMDb [8]. Half of them are positive and the other half are negative.

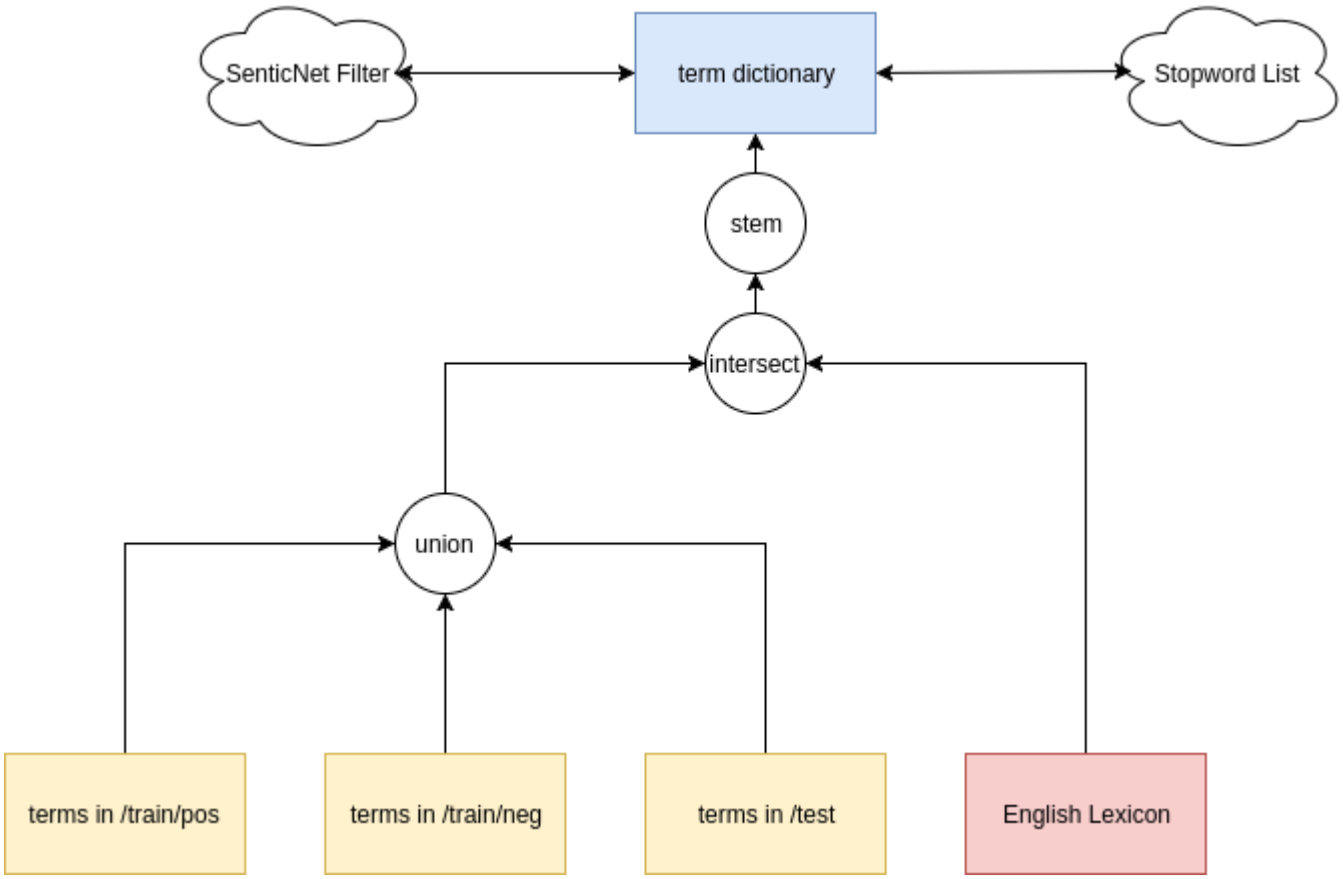


Figure 3.1: Depiction of the diagram of the lexicon.

Preprocessing process

- Punctuation removal, tokenization.
- Stemming (non-English terms are deleted).
- Terms frequencies.
- TF/IDF Vectors.

Prediction

The model uses a SVM classifier with SGD tuning for prediction. The vectors are defined with TF: log normalized and IDF: smooth normalized. The output is a text file that has in each row the name of the input text file and the output of the prediction for this file i.e 1.0 for positive and 0.0 for negative tweets.

Geoprediction Approach

1. Tokenization: Collect the tweets of every user and create a list of tokens for each user.

2. Stopwords removal.

3. Frequencies calculation: The most frequent word is given as a prediction. The algorithm also counts words that are included in hashtags (e.g paris in #prayforparis).

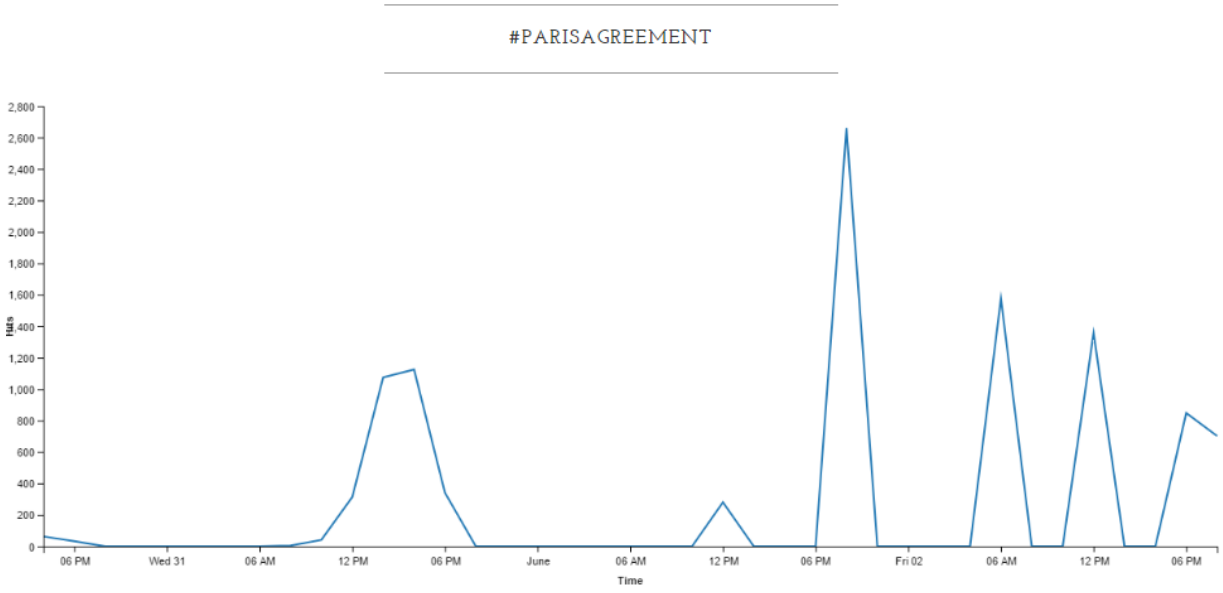
Event detection

Some top ranked terms (hashtags), their occurrences and the tweet that qualified as an event by our system will be presented in table 4.1

Term	Frequency
#parisagreement	10442
#covfefe	7085
#trumprussia	4229
#marchfortruth	973

Table 4.1: Terms and occurrences

A graph showing the top ranked term's occurrence over time can be seen in the following figure:



The tweet that our system determined as an event can be seen in the following figure:

Additional results can be found in the project's website [1].

Sentiment Analysis

The predictions of the sentiments of the tweets are presented using a pie chart in 4.1. The number of tweets that corresponds to each sentiment is presented in Table 4.2.

Opinion	Number of tweets
Negative	1501
Positive	1183
Total	2684

Table 4.2: Predictions of the tweets

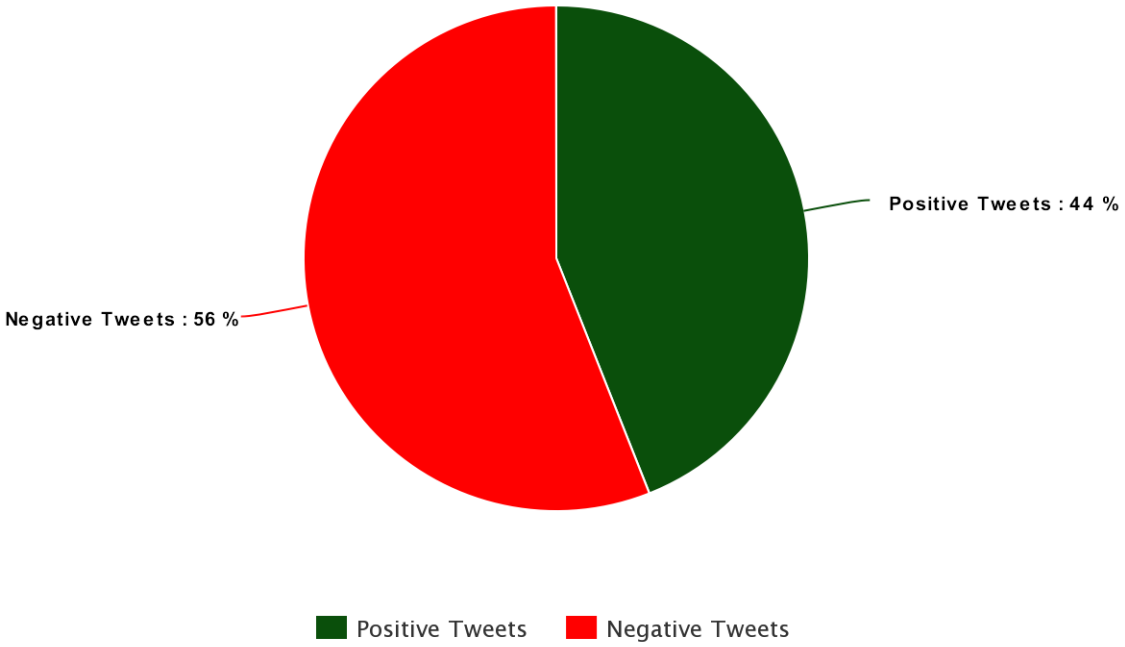


Figure 4.1: Percentages of negative and positive tweets.

Location Prediction

The chosen representation type for the results of Geoprediction experiments is the bar plot. The first chart represents the top referenced locations. The second one the top 20 locations for our experiments. The last bar chart depicts the accuracies of the models for every dataset. The metric used for the evaluation of our implementation is accuracy. The highest accuracy was achieved by the second dataset, in which the number of users and the number of cities is smaller (i.e 15 for cities and 15 for users/city) than in any other case.

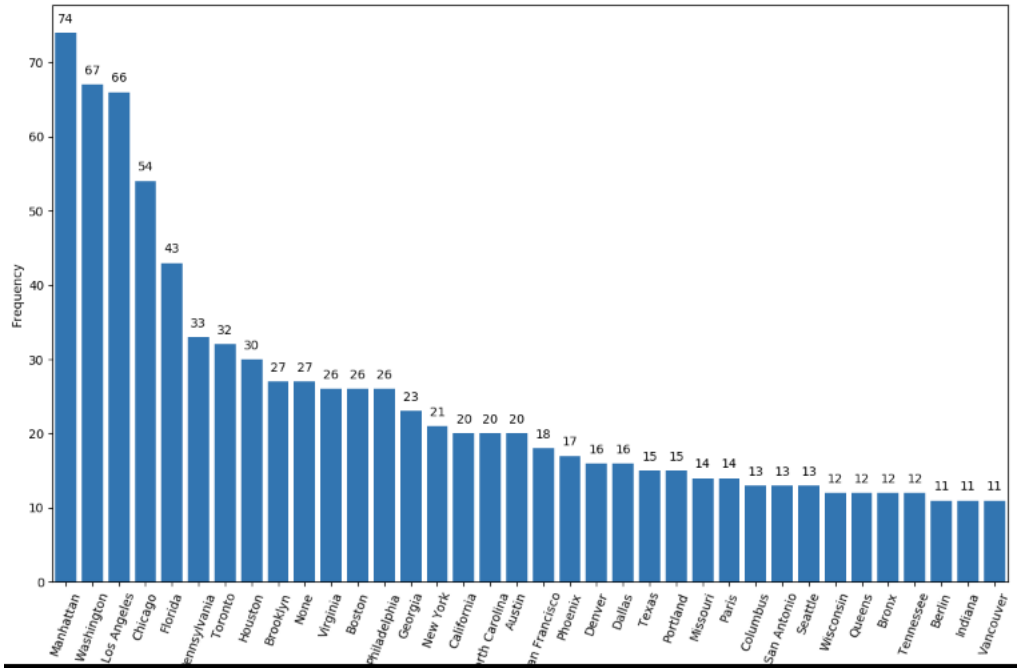


Figure 4.2: Most frequent locations

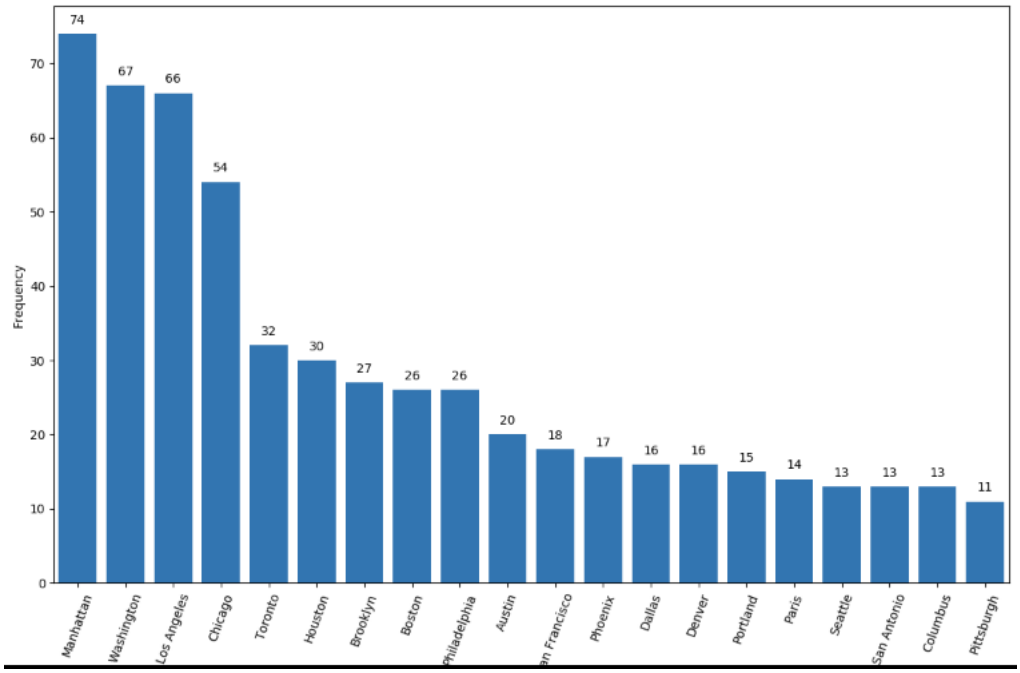


Figure 4.3: Top 20 selected locations

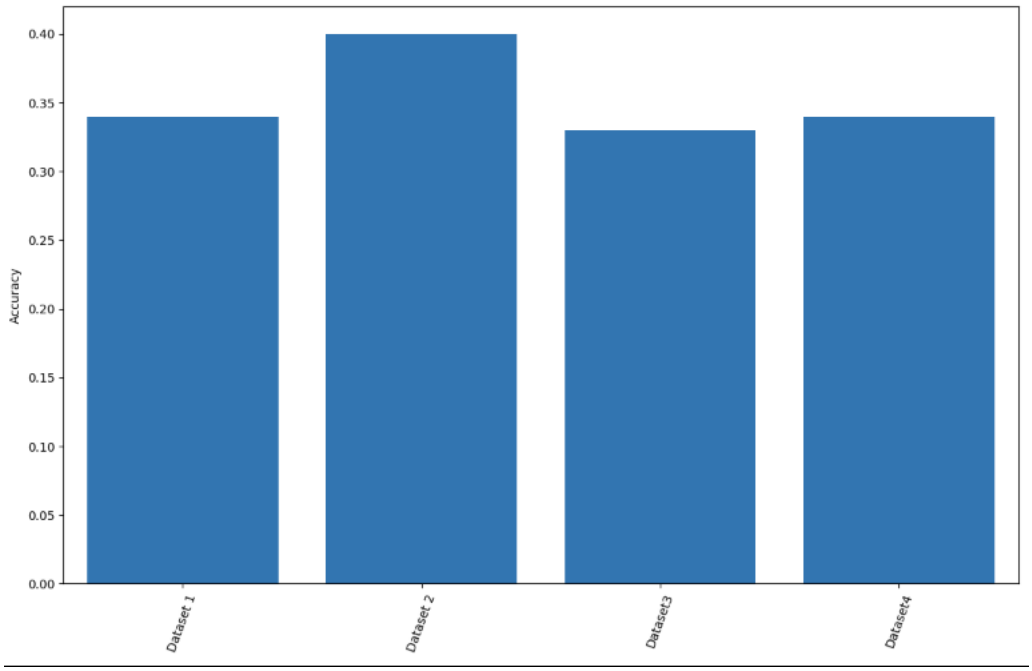


Figure 4.4: Accuracies of the datasets

Bibliography

[1] Website of the project. http://snf-755277.vm.okeanos.grnet.gr/.

[2] Project repository on github. https://github.com/alextsil/twitter-topic-detection-and-analysis/.

[3] Opinion mining tool. https://github.com/lefterisKl/Opinion-Mining/.

[4] Nltk. http://www.nltk.org/.

[5] pandas. http://pandas.pydata.org/.

[6] d3js. https://d3js.org/.

[7] Tweepy. http://www.tweepy.org/.

[8] Imdb. https://http://www.imdb.com/.

[9] M. Cataldi, L. D. Caro, and C. Schifanella. Personalized emerging topic detection based on a term aging model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):7, 2013.