# Twitter Sentiment Analysis of the Brexit result

**Surya Vajjhala**
U19590925
Email vajjhala@bu.edu

**Zhi Dou**
U21392913
Email zhidou@bu.edu

## Abstract

The main idea of the Project is to analyze twitter data pertaining to the "Brexit' vote and see if twitter could have predicted the result using sentiment analysis and supervised classification models . We also plan to understand the demographics of the voting patterns using clustering based on location. And finally to look at the fluctuation in peoples opinion over a period between March to September, looking at both before and after the deceleration of the result.

## Introduction

Britain's vote to leave the EU will have far reaching implications for the Britain, Europe and all countries doing trade with it. On June $23^{rd}$, UK voted to leave the EU, with 52% casting a vote to exit the European Union, which was perceived as a "shock" result among groups with the UK. Leading up the vote , many small events would have caused fluctuations in the opinion of the masses.

Twitter provides an active platform where people opinions can be tagged into certain categories with the help of the twitter "hashtags". Our aim is to analyze tweets pertaining to the "Brexit" , between the months of March to September and see if Twitter could have predicted the result of the vote.

Tweets will be classified as "Remain" or "Leave" based on the sentiment expressed in the text and the "hashtags" associated with it . We will then divide the data in to training and test data and use supervised classification algorithms to train a model.

## Dataset and Experiments

The Twitter API only provided the tweets in the past week ( past 7 days ). So to explore data before that time-period we had to manually scrape the data. For our previous experiments we scraped tweets in English with hashtag "#Brexit" and stored one tweet per 100 with attributes:

- ***geo:*** The location of the tweet.
- ***polarity:*** The measure of the sentiment of the tweet. It ranges from -1 to 1 where 1 is positive sentiment and -1 is negative sentiment.
- ***subjectivity:*** Measure of the subjectivity of the tweet, ranges form 0 to 1, where 0.0 is very objective and 1.0 is very subjective.
- ***time:*** The datetime object of the time when the tweet was posted.
- ***hashtags:*** The other hashtags associated with the tweet. These will be used to do further analysis and determining the content of the tweet.
- ***wordnouns:*** The likely important words in text.

Instead of storing "'text' itself, we implement sentiment analysis on the "text" once we downloaded it, and then we stored the result, which is the polarity and subjectivity into dataset. Finally we got tweets from March to July, 84371 tweets totally.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

## Initial Approach and Reason of Failure

Our initial approach had many shortcomings, we initially looked only at tweets with only #Brexit tag looked at only the sentiment associated with this tweet, if the polarity was negative then this tweet was categorized as "Leave" and if it was positive as "Remain" but that would invariably give a wrong classification. For example , take the tweet : " I hate the leave camp# Brexit ". This tweet would give a polarity of -0.8 and would be categorized as "Leave", though clearly it was " Remain". Hence we decided to use an alternative approach of looking at tags that would give a clear signal as to the opinion of the tweet based on the ten tags associated with each category that were mentioned above.

## Potential Improvement and Future Work

Instead of simply analyzing the sentiment of the sentences themselves, we should analyze the attitude of tweets towards Brexit.

To make tweets have more attitude, we will more key hashtags when we search tweets. The tags the were most associated withe the "Remain" camp are: #yes2eu, #yestoeu, #betteroffin, #votein, #ukineu, #bremain, #strongerin, #leadnotleave, #voteremain, #votein. And the tags most associated with the leave camp are: #no2eu, #notoeu, #betteroffout, #voteout, #britainout, #leaveeu, #loveeuropeleaveeu, #voteleave , #beleav.

Obviously we could get more information of attitude from this data. However in order to divided the tweets into two set, "support Brexit" and "against Brexit", we should associated these tags with the sentiment. If a tweet showed a positive polarity and had a tag associated with the "Remain" camp then it convincingly has a high probability to consider this tweets as "'support Brexit'. But if it contained negative polarity, the such a tweet was ignored. This same principal was applied for categorizing data into the "Leave" section as well.

For the new dataset following are the attributes are stored for analysis:

- ***text:*** The text present in the tweet.
- ***geo:*** The location of the tweet.
- ***location:*** Some tweets do not come with a location in which case we will use the location of the user to get the location.
- ***polarity:*** The measure of the sentiment of the tweet. It ranges from -1 to 1 where 1 is positive sentiment and -1 is negative sentiment.
- ***subjectivity:*** Measure of the subjectivity of the tweet, ranges form 0 to 1, where 0.0 is very objective and 1.0 is very subjective.
- ***time:*** The datetime object of the time when the tweet was posted.
- ***hashtags:*** The other hashtags associated with the tweet. These will be used to do further analysis and determining the content of the tweet.

After we get the classifier and classify all the data we have, for next step we want to find out what is the main topic in these two classes. The best way to find the main topic is a certain class is clustering these data, thus we will implement K-means and GMM to find different cluster inside class, to find the reason why people support or against Brexit.

## Techniques

After we get the training dataset, we need to train a classifier to classify a tweet is "support Brexit" or "against Brexit". We plan to build three classifier by using: Naive Bayes, Logistic Regression and Support Vector Machine.

### Naive Bayes

Naive Bayes is a simple and probabilistic classifier, which take input document and choose the class with maximum probability.

$$\hat{c} = argmax_{c \in C} P(c|d)$$

We need to consider word positions by simply go through all words in this text.

$$position \leftarrow \ all\ word\ positions\ in\ text\ document$$

$$c_{NB} = argmax P(c) \prod_{i \in positions} P(w_i|c)$$

However, there a some disadvantage to use Naive Bayes model. First assumption is: we assume position of words in text doesn?t matter. Second assumption is: these words are independently distributed with each other. In real world, these two assumption is hard to satisfied and also it impossible in real world. Thus we plan to try Logistic regression and SVM and compare these different classifier to get the best one.

## References

[1] github.com/Jefferson-Henrique/GetOldTweets-python.