

Sentiment Analysis of Election Result based on Twitter Data using R

Poli Ghosh¹, K.Vanitha²

¹P.G. Student, GITAM institute of science, GITAM University, Visakhapatnam 530045, India

²Assistant Professor, GITAM institute of science, GITAM University, Visakhapatnam 530045, India

Abstract – With the increase in the collection of texts obtained from various sources, the need to extract useful information from the texts in short span of time has also increased. To solve this problem, text mining is used. These information can further be used to provide better services such as setting trends, calculating sentiment analysis, graphical representation etc. Here, first the tweets of the people before as well as after the election are obtained from twitter. These tweets are then converted into text formats. These texts are then imported into R. They are then cleaned and made unified using various functions provided by text mining package in R[2]. Sentiment analysis is then calculated by comparing the positive and negative terms[1] which gives result. The text is then represented in the form of wordcloud where the keywords are depicted of the size according to their importance. An important keyword is represented with greater size as compared to other keywords. Representation of data in the form of Wordcloud is used because of quick visualization.

Key Words: Text Mining, R, Text import, Text cleaning, Sentiment analysis, Election result, Wordcloud.

1. INTRODUCTION

Users obtain texts in unstructured form from various sources such as financial and business reports, news articles, books, blog posts, messages or posts from social networking and other social media sites in ever increasing amounts. Therefore, there is a need to extract the information from these texts. In this paper, we calculate sentiment analysis of the election data collected from twitter before as well as after the election using R infrastructure. The results of the sentiment analysis will help us to understand people's opinion before and after the election. Greater the positive value of sentiment analysis, greater the positive attitude of the people. Similarly a negative value of the sentiment analysis shows a negative attitude and a zero value shows neutral attitude of the people. Firstly, the tweets of the people before and after the election are collected. These tweets are then converted into two text files, one containing the tweets before the election and the other after the election. Each of these text files are then imported into R. These files are then cleaned with the help of text mining package in R. Sentiment analysis is then calculated by comparing the positive and negative terms[1]. Lastly, each text document is represented by wordcloud. Wordcloud is the graphical representation of the text which gives the user a quick visualization of the important terms used in the text. Wordcloud is implemented by using the wordcloud package available in R.

2. IMPORT TEXT IN R

The text files containing few tweets of before as well as after the election are saved in the current working directory of R in .txt format. These texts are imported into R by using R tools[3].

```
> getwd()
```

This function is used to obtain the current working directory.

```
> readLines("beforeTrump.txt")
```

```
> readLines("afterTrump.txt")
```

This function is used to read line by line of the text files "beforeTrump.txt", "afterTrump.txt" of before and after election respectively.

2. CLEAN TEXT IN R

The above imported text files are cleaned[2] by applying various functions of R[3].

```
> text1 <- paste(readLines("beforeTrump.txt"), collapse=" ")
```

```
> text4 <- paste(readLines("afterTrump.txt"), collapse=" ")
```

The paste() function is used to make the entire content of the text file into a single line and then the result is assigned to variables text1 and text4.

```
> text2 <- gsub(pattern="\\W", replace=" ", text1)
```

```
> text5 <- gsub(pattern="\\W", replace=" ", text4)
```

The above script replaces all the punctuations with spaces from text1 and text4 and assigns the results in text2 and text5 respectively.

```
> text3 <- gsub(pattern="\\d", replace=" ", text2)
```

```
> text6 <- gsub(pattern="\\d", replace=" ", text5)
```

The above script replaces all the digits with spaces from text2 and text5 and assigns the results in text3 and text6 respectively.

```
> text3 <- tolower(text3)
```

```
> text6 <- tolower(text6)
```

The above script lowercases all the words in text3 and text6 and assigns the results in text3 and text6 respectively.

```
> install.packages("tm")
```

The above script install the text mining package named "tm".

```
> library(tm)
```


