ISSN: 2455-2631 © June 2016 IJSDR | Volume 1, Issue 6

Predicting the disease outbreaks using Social Network Enabled FLU Trends

PV Naveen Kumar[1], A. Haritha[2], Y.N.S. Ayyappa[3], G.Lakshmi[4]

[1,3]Research Scholar, [2,4]Assistant Professor
PVP Siddthartha Institute of Technology, VJA.

Abstract: **Data is being produced colossally on social media, a platform where people voice their experiences & opinions on varied contexts. Health Care being one of them, topics like health conditions, their symptoms, treatments, side effects, and so on will be discussed inevitably. This makes the publicly available social media data an invaluable resource for mining such data to discover interesting and actionable healthcare insights. The main objective of our project is to reduce the impact of seasonal influenza epidemics. We present the method, which monitors messages posted on Twitter with a mention of flu indicators to track and predict the emergence and spread of an influenza epidemic in a population. Based on the data collected during 2 weeks from 20th December 2015.The analysis results will subsequently reported visually in terms of, distribution of flu types, flu symptoms, and flu treatments. This method can be very useful for early prediction of flu outbreaks, which in turn can facilitate faster and better response preparation.**

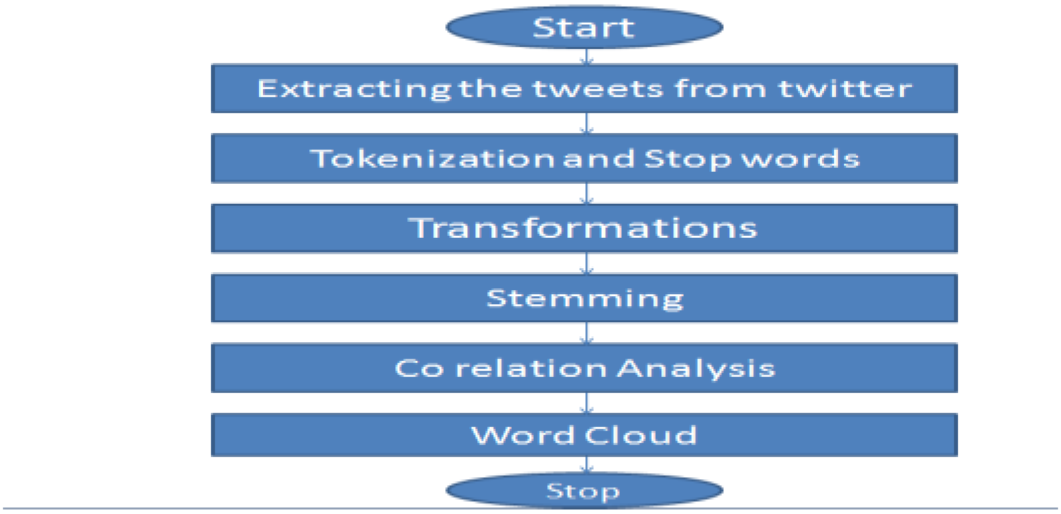
Key Words: **Tokenize and Stop words, Correlation Analysis, Word Cloud, and Twitter**

I.Introduction

The proposed system will mainly concentrate on social network enabled data. Twitter is a popular micro-blogging service where users can post short messages limited to 140 characters. Twitter has been used as a medium for real-time information dissemination and it has been used in various brand campaigns, elections, and as a news media. Because Twitter data can be collected in real-time, it has been used to predict real world outcomes.. Although a very high volume of twitter stream contains general chatter, it does contain enough health-related information to track disease spread.

II.Proposed System:

We proposed a method that uses twitter data to track influenza activities in real-time[1]. The proposed system consists of four stages: Data Collection, Data Preprocessing, Data modeling and Data Visualization[2]. The data collecting module continuously downloads flu-related public twitter data using Twitter streaming API[3]. The overall process is shown in the below figure:



The preprocessor module extracts tweet texts, and stores them in a database for further analysis[5]. In text modeling, we track and compare popularity of different flu types, symptoms, and treatments. In the final data visualization stage, the popularity of different flu types, symptoms, and treatments (the output of the text model) are presented as bar charts, for easy visualization and comparison.

a.Tokenize and Stopwords:

This will splits the text of a document into a sequence of tokens. We have to split the entire data sets into an individual tokens consisting of one single word, what's the most appropriate option before finally building the word vector. Or if you are going to build windows of tokens or something like that, you will probably split complete sentences[6]. As a result, each word in the text is represented by a single token. The stopwords, will filters English stopwords from a document by removing every token which equals a stopword from the built-in stopword list.

b.Transformations:

Once we have corpus we can modify the document. For example: stopwords removal, stemming etc.Transformations are done via **tm_map()** function which applies to all elements of corpus and all transformations can be done in single text documents

To clean the data file various commands are used which are listed below:

To convert to lower case:

corpus <- tm_map(corpus, content_transformer(tolower))

To remove punctuation:

corpus <- tm_map(corpus, removePunctuation)

To eliminating extra whitespaces

corpus <- tm_map(corpus, stripWhitespace)

c.Stemming:

The Porter stemming algorithm (or ‘Porter stemmer’) is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems. Stem porter Reduce words to their stems.

Porter stemmer stems "universal", "university", and "universe" to "univers". This is a case of over stemming: though these three words are etymologically related, their modern meanings are in widely different domains, so treating them as synonyms in a search engine will likely reduce the relevance of the search results.

d.Correlation Analysis:

A correlation is a number between -1 and +1 that measures the degree of association between two attributes (call them X and Y). A positive value for the correlation implies a positive association. In this case large values of X tend to be associated with large values of Y and small values of X tend to be associated with small values of Y. A negative value for the correlation implies a negative or inverse association. In this case large values of X tend to be associated with small values of Y and vice versa.

Suppose we have two attributes X and Y, with means X' and Y' respectively and standard deviations S(X) and S(Y) respectively. The correlation is computed as summation from 1 to n of the product *(X(i)-X').(Y(i)-Y')* and then dividing this summation by the product *(n-1).S(X).S(Y)* where *n* is total number of examples and *i* is the increment variable of summation. There can be other formulas and definitions but let us stick to this one for simplicity.

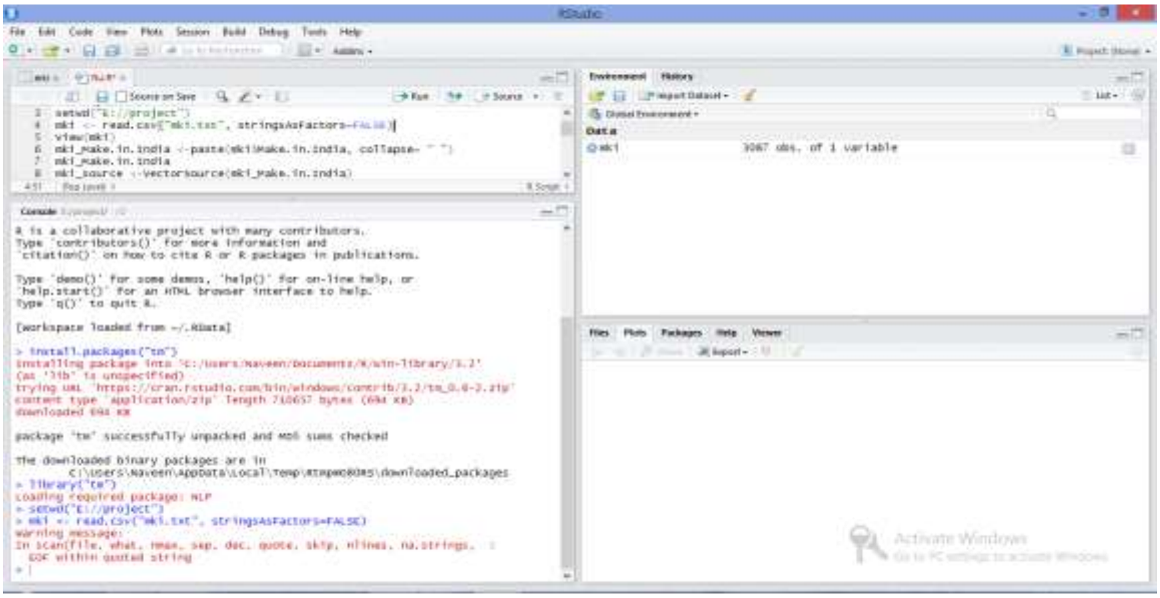
e.WordCloud:

After building a document term matrix and frequency terms.we can show the importance of words with a wordcloud.

The words cloud can be formed by using the following command:

wordcloud(words[1:100] ,frequency[1:100])

III. Results and Discussions

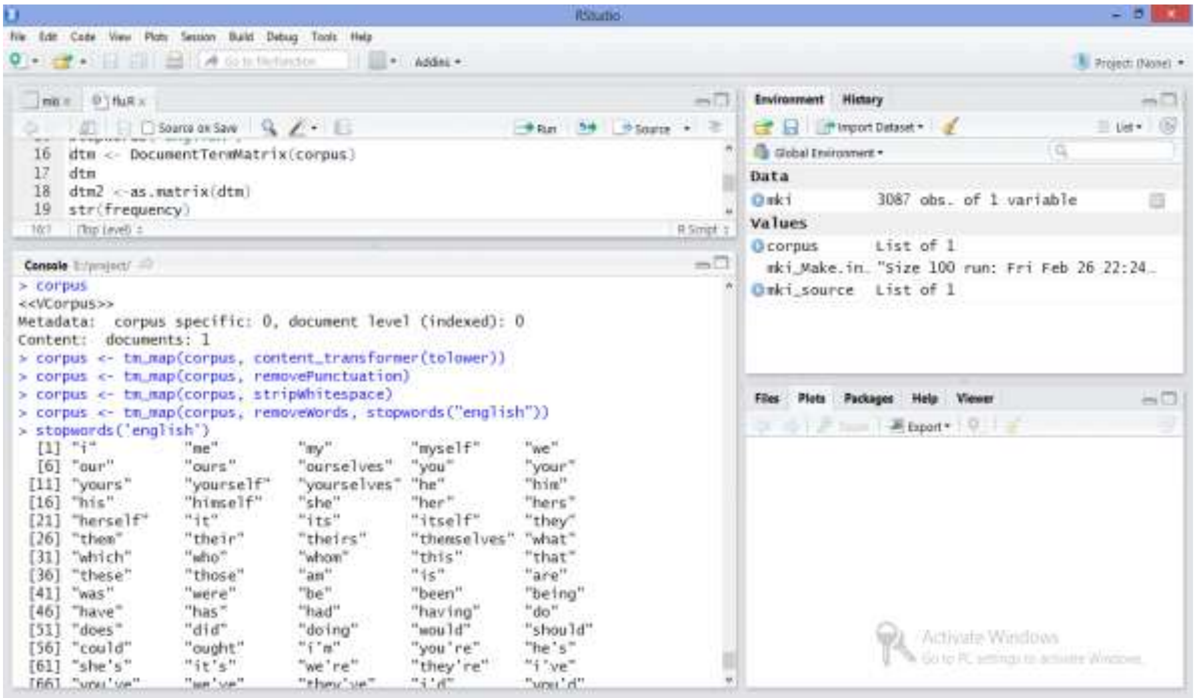


```
setwd("E://project")
mk1 <- read.csv("mk1.txt", stringsAsFactors=FALSE)
View(mk1)
mk1_make.in.india <-paste(mk1$Make.in.india, collapse=" ")
mk1_make.in.india
mk1_source <-VectorSource(mk1_make.in.india)
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> install.packages("tm")
Installing package into 'C:/Users/Naveen/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/tm_0.6-2.zip'
Content type 'application/zip' length 710657 bytes (694 KB)
downloaded 694 KB

package 'tm' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Naveen\AppData\Local\Temp\RtmpWC8DM5\downloaded_packages
> library("tm")
Loading required package: NLP
> setwd("E://project")
> mk1 <- read.csv("mk1.txt", stringsAsFactors=FALSE)
Warning message:
In scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings,  :
  EOF within quoted string
> |
```

```
Data
mk1                3067 obs. of 1 variable
```

Figure: Loading the Data into R console

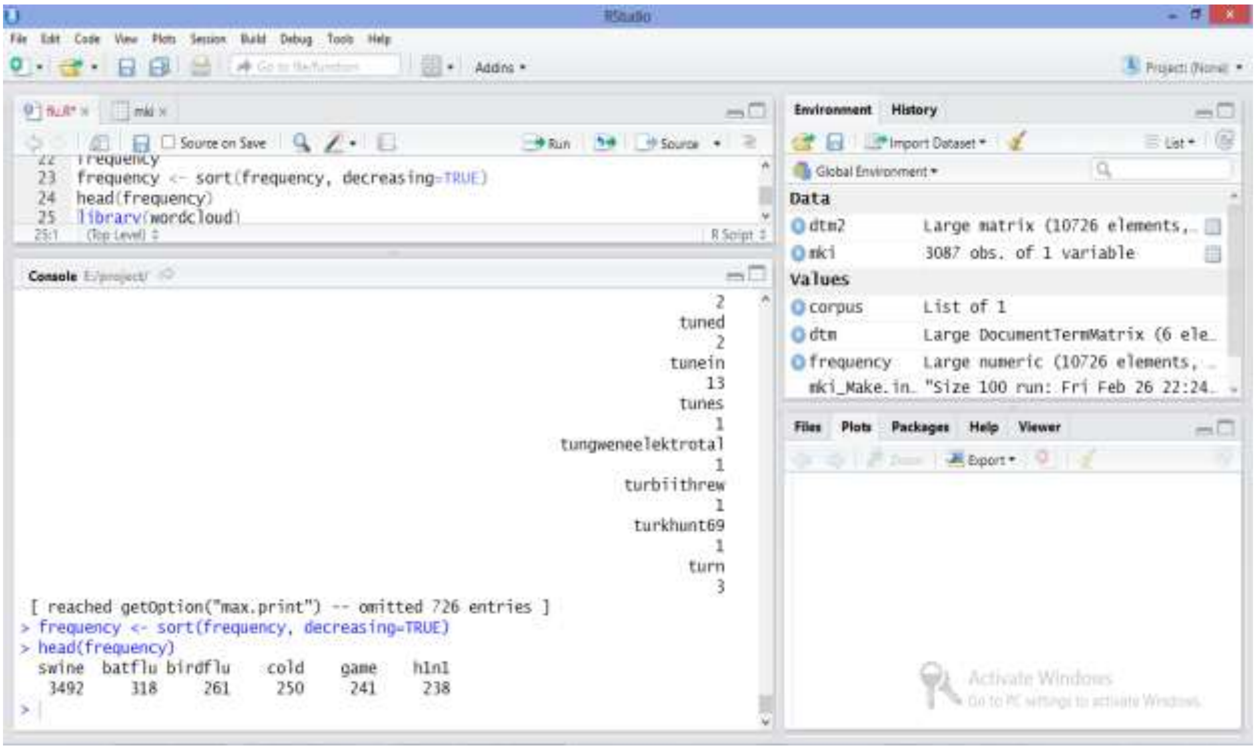


```
dtm <- DocumentTermMatrix(corpus)
dtm
dtm2 <-as.matrix(dtm)
str(frequency)
```

```
> corpus
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 1
> corpus <- tm_map(corpus, content_transformer(tolower))
> corpus <- tm_map(corpus, removePunctuation)
> corpus <- tm_map(corpus, stripWhitespace)
> corpus <- tm_map(corpus, removeWords, stopwords("english"))
> stopwords('english')
  [1] "i"          "me"         "my"         "myself"     "we"
  [6] "our"        "ours"       "ourselves"  "you"        "your"
 [11] "yours"      "yourself"   "yourselves" "he"         "him"
 [16] "his"        "himself"    "she"        "her"        "hers"
 [21] "herself"    "it"         "its"        "itself"     "they"
 [26] "them"       "their"      "theirs"     "themselves" "what"
 [31] "which"      "who"        "whom"       "this"       "that"
 [36] "these"      "those"      "am"         "is"         "are"
 [41] "was"        "were"       "be"         "been"       "being"
 [46] "have"       "has"        "had"        "having"     "do"
 [51] "does"       "did"        "doing"      "would"      "should"
 [56] "could"      "ought"      "i'm"        "you're"     "he's"
 [61] "she's"      "it's"       "we're"      "they're"    "i've"
 [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
```

```
Data
mki            3087 obs. of 1 variable
Values
corpus         List of 1
mki_Make.in…   "Size 100 run: Fri Feb 26 22:24…
mki_source     List of 1
```

Figure: Transforming the Data

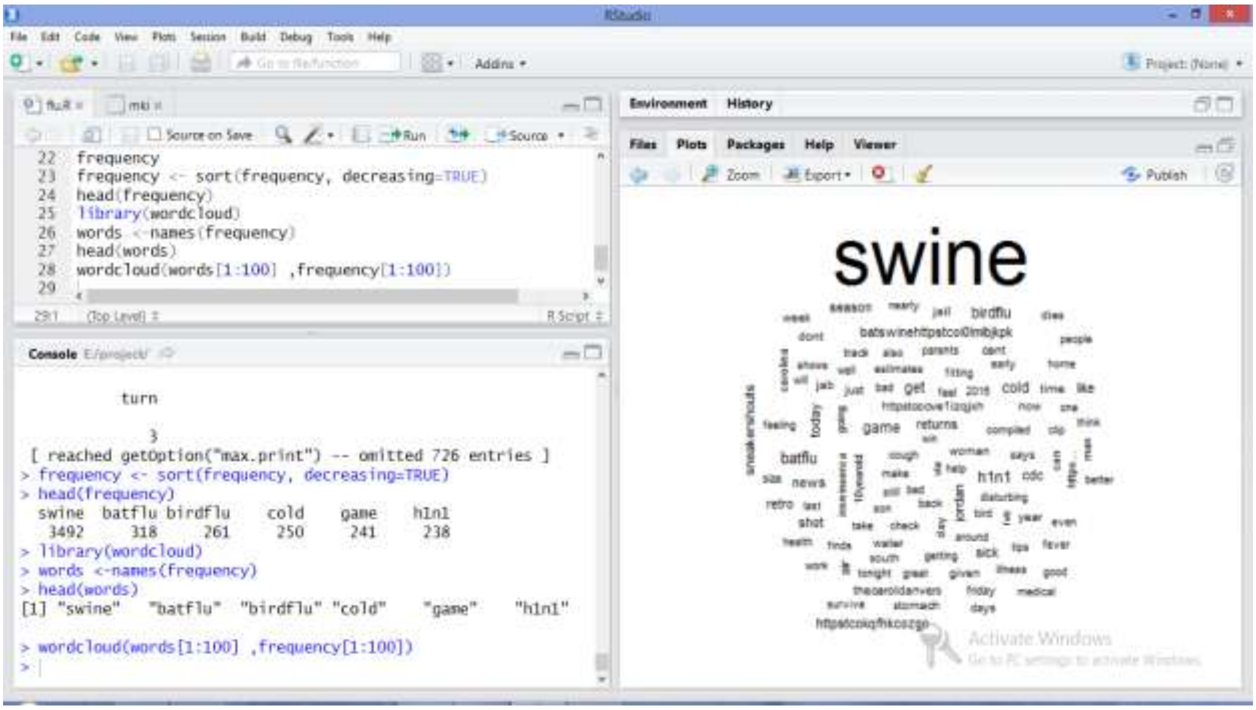


```
frequency
frequency <- sort(frequency, decreasing=TRUE)
head(frequency)
library(wordcloud)
```

```
                                                   2
                                               tuned
                                                   2
                                              tunein
                                                  13
                                               tunes
                                                   1
                                   tungweneelektrotal
                                                   1
                                          turbiithrew
                                                   1
                                          turkhunt69
                                                   1
                                                turn
                                                   3
 [ reached getOption("max.print") -- omitted 726 entries ]
> frequency <- sort(frequency, decreasing=TRUE)
> head(frequency)
 swine  batflu birdflu    cold    game    h1n1
  3492     318     261     250     241     238
>
```

```
Data
dtm2           Large matrix (10726 elements,…
mki            3087 obs. of 1 variable
Values
corpus         List of 1
dtm            Large DocumentTermMatrix (6 ele…
frequency      Large numeric (10726 elements, …
mki_Make.in…   "Size 100 run: Fri Feb 26 22:24…
```

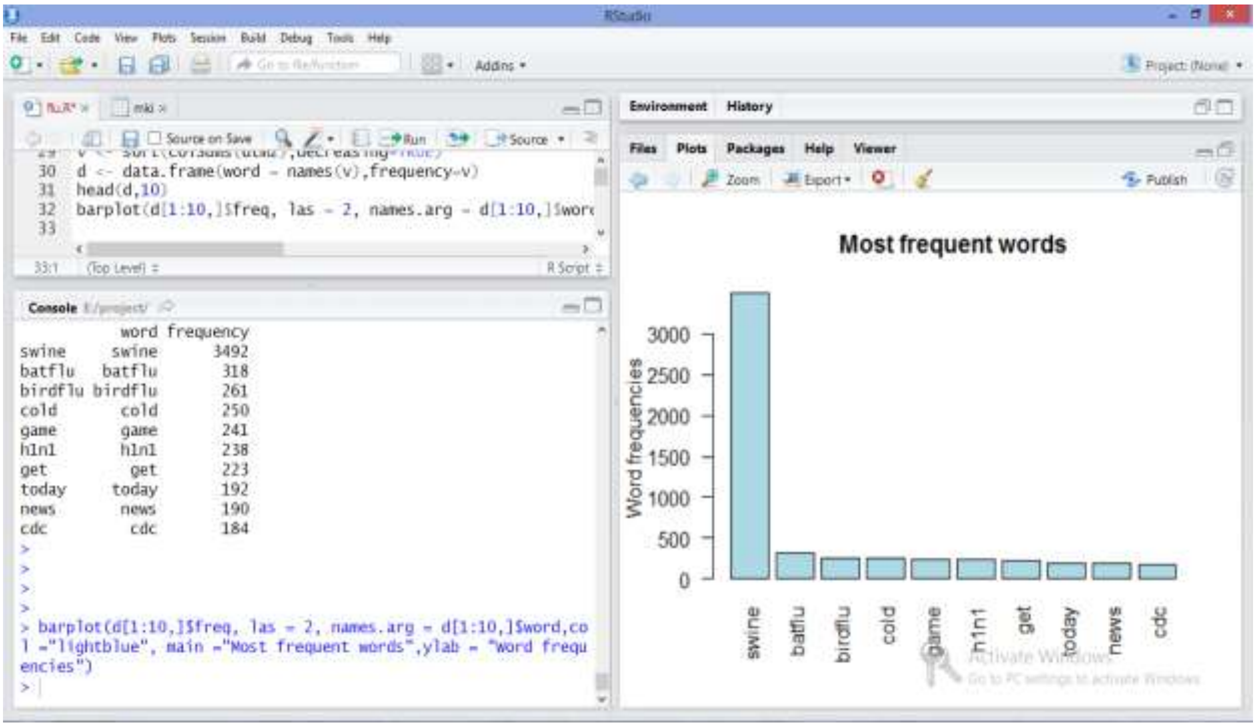
Figure :Frequency of individual terms and most Frequent terms



```
frequency
frequency <- sort(frequency, decreasing=TRUE)
head(frequency)
library(wordcloud)
words <-names(frequency)
head(words)
wordcloud(words[1:100] ,frequency[1:100])
```

```
          turn

             3
 [ reached getOption("max.print") -- omitted 726 entries ]
> frequency <- sort(frequency, decreasing=TRUE)
> head(frequency)
  swine  batflu birdflu    cold    game    h1n1
   3492     318     261     250     241     238
> library(wordcloud)
> words <-names(frequency)
> head(words)
[1] "swine"   "batflu"  "birdflu" "cold"    "game"    "h1n1"

> wordcloud(words[1:100] ,frequency[1:100])
> |
```

Figure: Building a word cloud with different types of flu’s and their symptoms



```
d <- data.frame(word = names(v),frequency=v)
head(d,10)
barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$wor
```

```
           word frequency
swine     swine      3492
batflu   batflu       318
birdflu birdflu       261
cold       cold       250
game       game       241
h1n1       h1n1       238
get         get       223
today     today       192
news       news       190
cdc         cdc       184
>
>
>
>
> barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,co
l ="lightblue", main ="Most frequent words",ylab = "Word frequ
encies")
> |
```

Figure: Bar plot for different types flu’s

Conclusion:

Twitter is quite well-known for its real-time nature. Tweets often have the indications of an occurrence & when keenly observed, we can draw some useful deductions from them. On the other hand, most of the diseases or ailments can be treated & cured well when they are identified in their early stages; and that identification is possible with the vigilant study of the symptoms. Therefore, using the tweets to identify the words that may mean the symptoms of the flu helps in identifying the flu on the whole. This methodology thus is very useful for predicting the type of flu and suggesting the right treatment for the users.

References:

[1] F. Jordans, “WHO working on formulas to model swine flu spread,” 2009.

[2] N. M. Ferguson, D. A. Cummings, S. Cauchemez, C. Fraser, S. Riley, A. Meeyai, S. Iamsirithaworn, and D. S. Burke, “Strategies for containing an emerging influenza pandemic in southeast asia,” Nature, vol. 437, 2005.

[3] I. Longini, A. Nizam, S. Xu, K. Ungchusak, W. Hanshaoworakul, D. Cummings, and M. Halloran, “Containing pandemic influenza at the source,” Science, vol. 309, no. 5737, 2005.

[4] J. Espino, W. Hogan, and M. Wagner, “Telephone triage: A timely data source for surveillance of influenza-like diseases.” in AMIA: Annual Symposium Proceedings, 2003.

[5] Haritha Akkineni, P.V.S Lakshmi, B. Vijay Babu, G.Lakshmi ,Modeling and Visualizing the Extraction of Opinions from Twitter, International Journal of Innovations & Advancement in Computer Science, IJIACS Volume 5, Issue 2 February 2016.

[6] Haritha Akkineni, P.V.S Lakshmi, B. Vijay Babu Online Crowds Opinion-Mining it to Analyze Current Trend: A Review, International Journal of Electrical and Computer Engineering(IJECE), Vol 5, No 5, OCT, 2015, IJECE,ISSN: 2088-8708, SCOPUS indexed Journal)(IAES).