Sentiment Analysis of Harvey twitter with R

Vinh The Nguyen

[1]Computer Science Department
Texas Tech University

vinh.nguyen@ttu.edu

Abstract. *Sentiment analysis or opinion mining is an emerging trend that uses of natural language processing to extract, identify quantify, and study subjective information systematically. It has been widely used in social network, marketing where opinions of a particular phenomenon play an important role. Last month, we witnessed a big hurricane Harvey with a lot of damage on people and proper- ties. It is important to understand the opinion of the hurricane Harvey from different perspectives in order to better prepare for the future events. In this project, our team will mining opinion on the social network, particularly on Twitter. Data will be retrieved through Twitter Search API. Pre-processing steps include removing stop-words, punctuation, white space, and non-english characters. After data is clean, several methods will be used to mining the corpus such as Association rules mining, Latent Dirichlet and Wordcloud. The expected outcome of this project is to understand people's opinion toward hurricane Harvey. The project will be implemented in R.*

1. Data Retrieval

Twitter provides a great API that allows registered user to retrieve a number of tweets within a given period (usually 7 days). It is required that an application has to be created in the user's account to get key and token (via http://apps.twitter.com). Essentially, a minimum of 4 pieces of information are required: consumer key, consumer secret, access key, and access secret. We use TweeteR package to pull tweets with our desired keywords: Tropical Depression Harvey. The use of the TweeteR package is shown below.

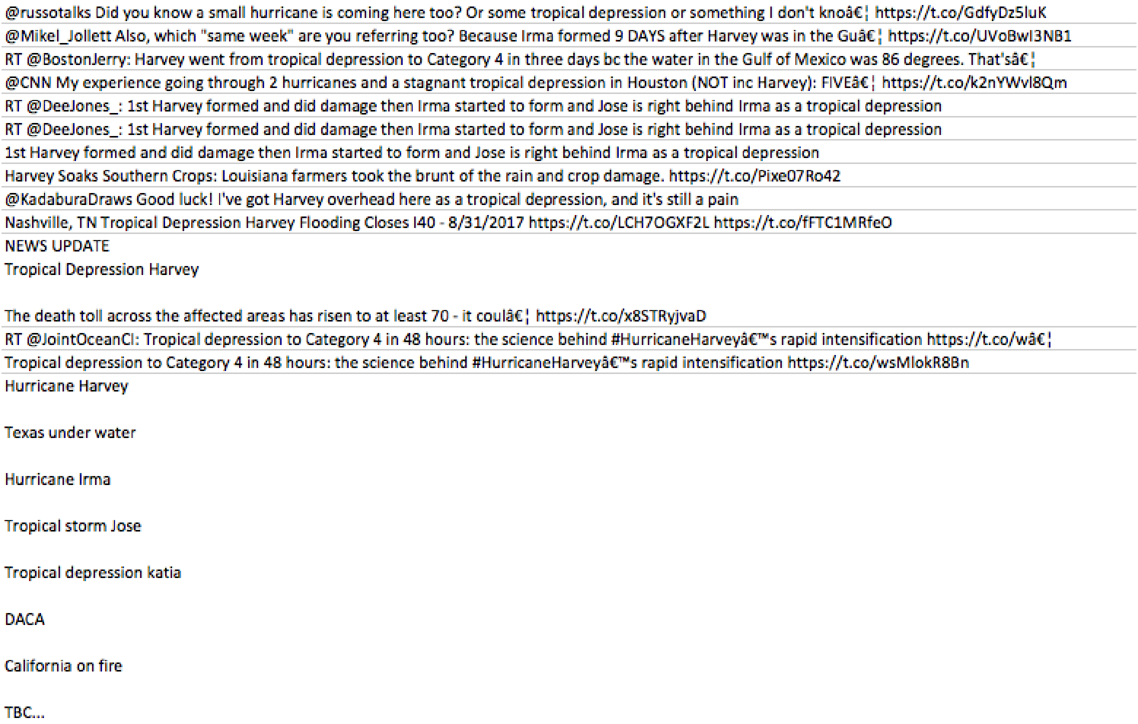
```
1 install.packages('TweeteR')
2 require('TweeteR')
3 setup_twitter_oauth(consumer_key,consume_secret,access_key,
     access_secret)
4 tweets <- searchTwitter(searchString= 'Tropical + Depression +
     Harvey', n=10000, lang='en', since='2017-08-26', until='
     2017-09-08')
5 tweetds <-twListToDF(tweets) #Convert data to data frame
6 write.csv(tweetds, "TwitterData.csv") # export data frame to csv
     file
```

The first three lines (1–3) install the TweeteR package, import it to current session and setup authentication to Twitter API. Line 4 performs a search function on Twitter with the following parameters:

- *searchString*: is the keywords issue to twitter, we use "+" to separate keyword terms.This parameter is required.
- *n*: is the maximum number of tweets to return.

- *lang*: is the language of tweets, we are only interested in english tweets.
- *since*: is the start time to collect the tweets.
- *until*: is the end time to collect the tweets.

The function *searchTwitter* return any authorized tweets that match the search criteria. It is noted that it is often not get as many as n argument tweets because of pagination restriction. Our search result returned 5,312 tweets. Line 5–6 simply converts the search results into data frame and write to disk as a Comma Separated Value file. Figure 1 illustrates a portion of twitter data that we retrieved.



```
@russotalks Did you know a small hurricane is coming here too? Or some tropical depression or something I don't knoâ€¦ https://t.co/GdfyDz5luK
@Mikel_Jollett Also, which "same week" are you referring too? Because Irma formed 9 DAYS after Harvey was in the Guâ€¦ https://t.co/UVoBwI3NB1
RT @BostonJerry: Harvey went from tropical depression to Category 4 in three days bc the water in the Gulf of Mexico was 86 degrees. That'sâ€¦
@CNN My experience going through 2 hurricanes and a stagnant tropical depression in Houston (NOT inc Harvey): FIVEâ€¦ https://t.co/k2nYWvl8Qm
RT @DeeJones_: 1st Harvey formed and did damage then Irma started to form and Jose is right behind Irma as a tropical depression
RT @DeeJones_: 1st Harvey formed and did damage then Irma started to form and Jose is right behind Irma as a tropical depression
1st Harvey formed and did damage then Irma started to form and Jose is right behind Irma as a tropical depression
Harvey Soaks Southern Crops: Louisiana farmers took the brunt of the rain and crop damage. https://t.co/Pixe07Ro42
@KadaburaDraws Good luck! I've got Harvey overhead here as a tropical depression, and it's still a pain
Nashville, TN Tropical Depression Harvey Flooding Closes I40 - 8/31/2017 https://t.co/LCH7OGXF2L https://t.co/fFTC1MRfeO
NEWS UPDATE
Tropical Depression Harvey

The death toll across the affected areas has risen to at least 70 - it coulâ€¦ https://t.co/x8STRyjvaD
RT @JointOceanCI: Tropical depression to Category 4 in 48 hours: the science behind #HurricaneHarveyâ€™s rapid intensification https://t.co/wâ€¦
Tropical depression to Category 4 in 48 hours: the science behind #HurricaneHarveyâ€™s rapid intensification https://t.co/wsMlokR8Bn
Hurricane Harvey

Texas under water

Hurricane Irma

Tropical storm Jose

Tropical depression katia

DACA

California on fire

TBC...
```

Figure 1. Example of raw twitter data

2. Data Cleaning

As depicted in Figure 1, our data contains a lot of noise that does not contribute too much in our analysis, it is reasonable to remove this noise such as non-asci characters (line 1), url (line 3), tagging people (line 4), none-english characters (line 5), new line (line 6), stopword (line 7), white space (line 8), or even duplication (line 10). Some words have many forms (singular, plural, past tense..) but they have the same meaning, so we stem these words (line 9). When removing stopwords, it is noticed that some words that are not in the dictionary, these words maybe abbreviation or refer to something that we don't know, we also remove these words. In addition, the three words (harvey, tropical, depression) are the keywords that we input into the search function, so it is anticipated that these words will appear in almost tweets. For better analysis, these words should also be taken out. Our cleaned data is saved into $twitter$ (line 10)

```
1  tweets <- iconv(tweets, "latin1", "ASCII", sub="")
2  tweets <- tolower(tweets)
3  tweets <- gsub("http[^[:space:]]*", "", tweets)
```

```
tweets <- gsub("@[^[:space:]]*", "", tweets)
tweets <- gsub("[^[:alpha:][:space:]]*", "", tweets)
tweets <- gsub("\n", " ", tweets)
tweets <- removeWords(tweets,c(stopwords('english'), c("tropic","depress","harvey","al","ion","gu","kno","amp","ap","ne","w","via","us","near","now","rt","will")))
tweets <- stripWhitespace(tweets)
tweets <- stemDocument(tweets)
twitter <- unique(tweets)
```

3. Methods

In order to mining opinion over social network, some interesting questions should be addressed such as: What are people mostly talking about this event? Is there any relationship or association between these talks? Is there any common viewpoint? Can we capture most of the important incidents in this data. To answer these posing questions, we adopt three methods: Association rule mining technique to find out the relationship among the talks, topic modeling to discover common viewpoint and word cloud for a picture worth thousand of words.

3.1. Association rule mining

Association rule mining is a rule-based machine learning technique, its purpose is to find interesting hidden relations among variables in the dataset. For example, in Market Basket Analysis, this method is used to find frequent items set often bought together so sellers can arrange those items in a convenient way for customers or give recommendations. Table 1 illustrates this example with extracted 5 transactions. We would like to know the likely that, for example, if a customer buy Milk and Diaper, will he likely buy a beer? Association rule mining allows us to unveil this type of questions.

Table 1. Example of 5 transactions in a store

Transaction ID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Diaper, Milk, Beer
5	Bread, Diaper, Milk, Coke

To measure the association, there are three indicators:

- *Support*: This measure indicates the popularity of an itemset, which is the proportion of transactions that contain itemset. For example, in Table 1 the support of {Bread} is 4 out of 5, or 80 %. Itemsets can also contain multiple items. For instance, the support of {Bread, Milk} is 3 out of 5, or 60%. If you find out that sales of some items beyond a certain proportion tend to give big profits, this certain proportion can be considered as a support threshold then it can identify itemsets with support values above this threshold as significant itemsets.
- *Confidence*: This measure says how likely an item Y will be bought when item X is purchased, it is expressed as $\{X\} \rightarrow \{Y\}$. This is calculated by:

$Confidence = \frac{\{X,Y\}.count}{\{X\}.count}$
Where $\{X, Y\}.count$ is the number of transactions that contain both X and Y, and $\{X\}.count$ is the number of transactions that contain X.
For example, $Confidence$ of the transaction $\{Milk, Diaper\} \to \{Beer\}$ is $\frac{2}{3} = 0.667$.

- *Lift*: Similar to Confidence measure, Lift also says how likely an item Y will be bought when item X is purchased while controlling the popularity of Y. It is expressed as: $Lift = \frac{\{X,Y\}.count}{\{X\}.count * \{Y\}.count}$
If the value of Lift is 1, meaning that there is no association between items. If Lift is greater than 1, it means the item Y is likely to be bought if item X is bought. If Lift is less than 1, item Y is unlikely to be bought if item X is bought.

3.2. Topic modeling (Latent Dirichlet Allocation)

Latent Dirichlet Allocation is very popular topic modeling technique that allows users to organize, search and summarize a large corpus of information. The basic idea of this approach is that it considers each word in a document belongs to one of the document's topics. For example, we have the following documents

- I like to eat broccoli and bananas
- I ate a banana and spinach smoothie for breakfast
- Chinchillas and kittens are cute
- My sister adopted a kitten yesterday
- Look at this cute hamster munching on a piece of broccoli

The LDA model discovers that document 1,2,3,5 contains topic 1 and topic 2 and document 4 contains only topic 2. The procedure of creating and assigning topic is as follow:

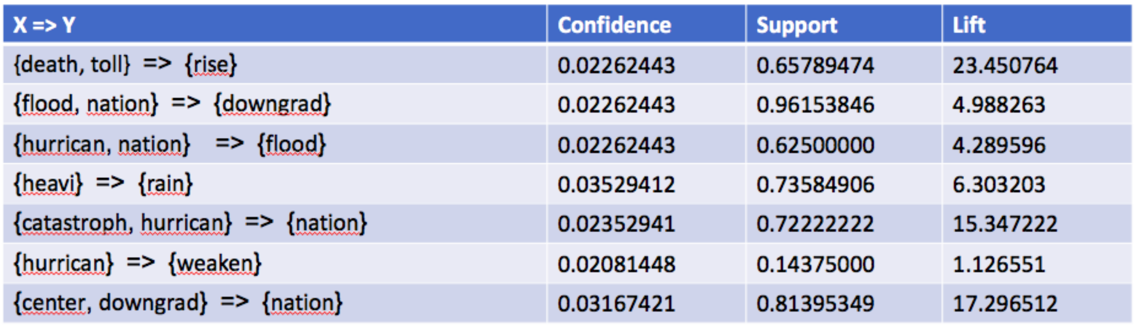
- Step 1: Pre-select the number of K topics
- Step 2: Go through each document and randomly assign each word to one of K topics
- Step 3: For each document d, go through each word w then calculate
$p\_z = P1 * P2$.
Where P1 is the p(word w — topic t) = the proportion of assignments to topic t over all documents that come from this word w. And P2 is p(topic t — document d) = the proportion of words in document d that are currently assigned to topic t. Re-assign word w with new topic t with the probability p_z. Repeat Step 3 with the number of given iteration.

3.3. Word Cloud

Word cloud or tag cloud is one one the most common ways to visualize the popularity of a word in terms of its size and color. The bigger the word, the more important. The layout algorithm itself is incredibly simple. For each word, starting with the most "important":

- Attempt to place the word at some starting point: usually near the middle, or somewhere on a central horizontal line.
- If the word intersects with any previously-placed words, move it one step along an increasing spiral. Repeat until no intersections are found.

Font size is linearly scaled corresponding to its frequency.



X => Y	Confidence	Support	Lift
{death, toll} => {rise}	0.02262443	0.65789474	23.450764
{flood, nation} => {downgrad}	0.02262443	0.96153846	4.988263
{hurrican, nation} => {flood}	0.02262443	0.62500000	4.289596
{heavi} => {rain}	0.03529412	0.73584906	6.303203
{catastroph, hurrican} => {nation}	0.02352941	0.72222222	15.347222
{hurrican} => {weaken}	0.02081448	0.14375000	1.126551
{center, downgrad} => {nation}	0.03167421	0.81395349	17.296512

Figure 2. Example of some association rules

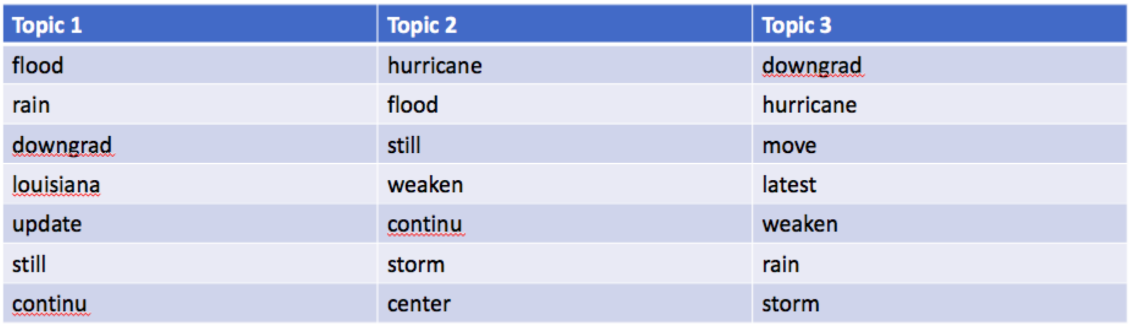
4. Result

4.1. Association Rule Mining

Figure 2 illustrates some useful rules extracted from the model. Because our data is so sparse and we remove all retweet and duplication, it is more challenging to 'connect' the words with high confidence and support. To find some grain in the data, we lower the threshold to 0.02 and we find some interesting association. For example, $(flood, nation) \rightarrow (downgrad)$ or $(hurricane) \rightarrow (weaken)$. These links can somehow be predictable because the time frame for collecting data is nearly at the end of the storm. But another interesting pattern is $(death, toll) \rightarrow (rise)$ with very high Lift measure value. This automatically finding pattern plays an important role in national rescue.

4.2. Topic modeling

Machine learning is an art, deciding the number of topics and run the model does not always give desired results. In practice, we have to run the model several times with different values of K and select the wanted topics. Figure 3 extracted the most 7 popular terms in three topics (k=3) From these topics, we can infer the social opinion from the internet that the hurricane is getting weak in the center of texas and move to lousiana. This summarizing text is critical especially the rescue team has to plan within a short time.



Topic 1	Topic 2	Topic 3
flood	hurricane	downgrad
rain	flood	hurricane
downgrad	still	move
louisiana	weaken	latest
update	continu	weaken
still	storm	rain
continu	center	storm

Figure 3. Example of some topic modeling

4.3. Word cloud

Figure 4 provides an overview of the word cloud, it can be seen that the most mentioned word is "downgrad", accouting the most space on the chart. Storm, hurricane, flood or weaken are easily to spot, this quick overview of top mentioned words provides a positive sign of harvey storm which will end soon.

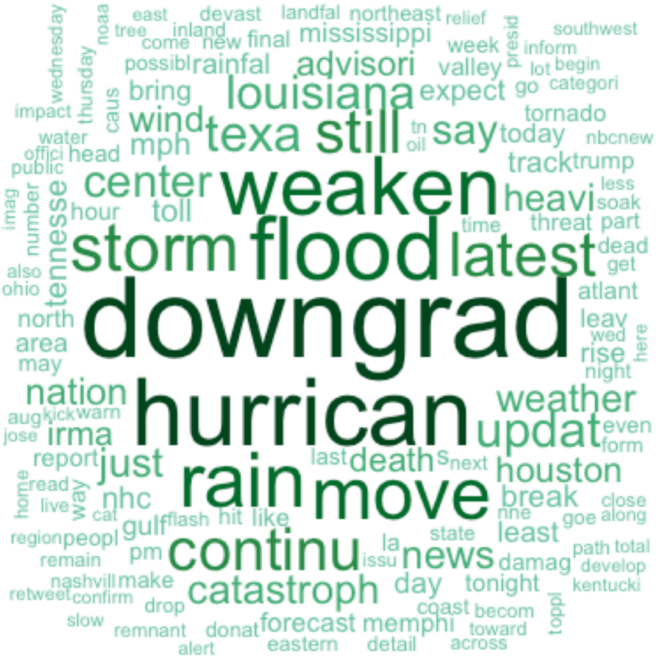


Figure 4. Example of some topic modeling

References