Who Made the News? Text Analysis using R

In this post, we perform detailed text analysis using headlines from different media and news sites, collected between March 2014 and August 2014.

This post covers the following tasks using R programming:
a) cleans the texts,
b) sorts and aggregates by publisher names
c) creates word clouds and word associations

The dataset used for the analysis was obtained from Kaggle Datasets, and is attributed to UCI Machine Learning. (Please see acknowledgement at the end). The raw tabular data includes information about news category (business, science and technology, entertainment, etc.)

R language has some useful packages for text pre-processing and natural language processing. Some of these packages we use for our analysis include:

- Wordcloud,
- qdap,
- tm,
- stringr,
- SnowballC

Analysis Procedure:

1. Read the source file containing text for analysis. I prefer fread() over read.csv() due to its speed even with large datasets.
   text_dict_source <- data.frame(fread("uci-news-aggregator.csv"))

2. For the scope of this program, we limit ourselves to only the headline text and publisher name.
   text_sourcedf <- subset(text_dict_source, select = c("ID", "TITLE", "PUBLISHER"))

3. Clean up the headlines by removing special characters and “emojis” from tweets, if any.

   text_sourcedf$TITLE <- sapply(text_sourcedf$TITLE,function(row) iconv(row, "latin1", "ASCII", sub=""))

4. Aggregate content from the top 20 Publisher. (in order of decreasing frequency) . Table alongside shows the Top 7.

x = data.frame(table(text_sourcedf$publ), stringsAsFactors = FALSE)

x = x[order(x$Freq, decreasing=TRUE),]

pubrct = 20 # using a variable to select number of publishers.

publ = x[1:(pubrct,]

PublisherName	Count
reuters	3902
huffington post	2455
businessweek	2395
contactmusic.com	2334
daily mail	2254
nasdaq	2228
examiner.com	2085

5. We use a “for loop” to filter and a custom function aggregate the headline texts for each publisher. (Code not shown)

6. Create word corpus with one object for each publisher. Clean and pre-process the text by removing punctuations, removing “stop words” (a, the, and, …) using tm_map() function as shown below:

   wordCorpus <- tm_map(wordCorpus, removePunctuation)

7. Create wordclouds for Publisher = "Reuters”. This can be done for any (or all) publishers using the wordcloud() function. The “color” option allows us to specify color palette, “rot.per” allows us to customize the word rotations and “scale” specifies whether word size should be dependent on frequency of occurrence.

   wordcloud(wordCorpus1, scale=c(5,0.5), max.words=100, random.order=FALSE,

   rot.per=0.35, use.r.layout=FALSE, colors = brewer.pal(16, "Dark2"))

We create wordclouds for 2 more publishers (Celebrity Café & CBS_Local) as shown below. Notice how the words for a financial news engine like Reuters differ from celebrity names and events (image2). Contrast them both with the everyday words (music, health, office) on a local news site like CBS (image 3).

Figure 1 Wordcloud – Reuters

Figure 2Wordcloud - CBS Local

Figure 3 Wordcloud - Entertainment Cafe

8. Word frequency and word associations. Some examples are given below:
   a. findAssocs(dtm, c("motorola"), corlimit=0.98)

```
> findAssocs(dtm, c("motorola"), corlimit=0.85)
$motorola
     shire      abbvi       surg      widen    pinnacl     profit
      0.96       0.94       0.94       0.93       0.92       0.92
      sale      estim   european    rebound     kodiak     confid
      0.92       0.91       0.91       0.91       0.90       0.89
  hillshir       slow blackberri      climb    eaccess       rise
      0.89       0.89       0.88       0.88       0.88       0.88
  treasuri     advanc    easyjet     german    guidanc        ibm
      0.88       0.87       0.87       0.87       0.87       0.87
    margin      oracl     output      slide      stock      trade
      0.87       0.87       0.87       0.87       0.87       0.87
    unilev     weaken     declin    generic     gilead     growth
      0.87       0.87       0.86       0.86       0.86       0.86
      loss  lululemon     mersch     draghi    sarepta      suiss
      0.86       0.86       0.86       0.85       0.85       0.85
```

Figure 4 Word Association

b. findAssocs(dtm, " ukrain"), corlimit=0.95) # specifying a correlation limit of 0.90

```
$ukrain
     bank       ecb     libya    nikkei   tension argentina   biotech
     0.98      0.98      0.98      0.98      0.98      0.97      0.97
   canada   concern    export    invest   condens      debt       eas
     0.97      0.97      0.97      0.97      0.96      0.96      0.96
    franc      fuel   guillon      kiwi    lafarg  mulberri peripheri
     0.96      0.96      0.96      0.96      0.96      0.96      0.96
   polici      roch    russia     spain     swiss    action     await
     0.96      0.96      0.96      0.96      0.96      0.95      0.95
    crisi      fiat    measur    mediat
     0.95      0.95      0.95      0.95
```

Figure 5 Word Association2

9. Plot word association:

   plot(dtm, term = xfrqdf, corThreshold = 0.12, weighting = F, attrs=list(node=list(width=20, fontsize=24, fontcolor="blue", color="red")))

The image (below) looks more like a plate of spaghetti, indicating all the words are inter-related. (logically correct since all the terms relate to global finance.)

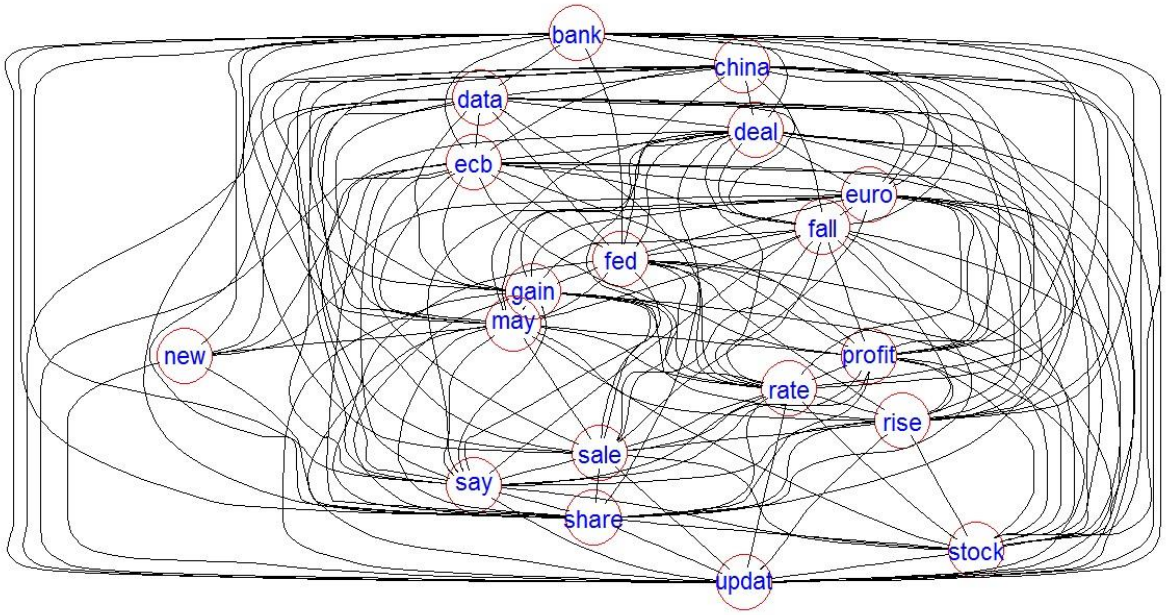

Figure 6 Word relationships for high-frequency words

The complete code for this analysis is available in the zipped folder, in both text and R program formats. Please do take a look and share your thoughts and feedback.

Acknowledgments:

This dataset comes from the UCI Machine Learning Repository.

Source - Lichman, M. (2013. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.