Analysing and Visualising Facebook Data

Suchandra Mandal and Ashfaque Azad

23 March 2018

Introduction

For analysis of facebook posts on galway, We have used two facebook pages as my source, namely Connacht tribune and Galway Today. We have extracted data for last 6 months in order to look at what news has been trending in Galway for the last few months. Two news sources were used to provide better weighting to news related to storms like ophelia etc which occurred for only a week. Thus, having more frequency of those terms will give them a better tf-idf rank in the code later on.

For now, let us load the required libraries.

```
# Loading all the required libraries
library(Rfacebook)
```

```
## Warning: package 'Rfacebook' was built under R version 3.4.4
```

```
## Loading required package: httr
```

```
## Warning: package 'httr' was built under R version 3.4.4
```

```
## Loading required package: rjson
```

```
## Loading required package: httpuv
```

```
## Warning: package 'httpuv' was built under R version 3.4.4
```

```
##
## Attaching package: 'Rfacebook'
```

```
## The following object is masked from 'package:methods':
##
##     getGroup
```

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.4.3
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:httr':
##
##     content
```

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 3.4.4
```

```
library(RTextTools)
```

```
## Warning: package 'RTextTools' was built under R version 3.4.4
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 3.4.3
```

```
library(SnowballC)
```

```
##
## Attaching package: 'SnowballC'
```

```
## The following objects are masked from 'package:RTextTools':
##
##     getStemLanguages, wordStem
```

```
library(sentimentr)
```

```
## Warning: package 'sentimentr' was built under R version 3.4.4
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 3.4.4
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 3.4.3
```

```
## Loading required package: RColorBrewer
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.4.3
```

```
options(warn = -1)
```

1. Extracting Facebook Data

We have used the facebook graph api here to extract public facebook posts from the news pages of Connacht Tribune and Galway Today from 22nd Aug, 2017 to 22nd Mar, 2018.After getting the data from the two news sources, both the dataframes were bounded together by rows since the number of columns for each is the same.

Since token is private, we have provided the code we used to extract the data and downloaded that as a csv which we have loaded here.

```
#Ashfaque Azad
# token <- '*****'

#Getting data from public news pages:
# an upper limit of 10000 was used to get all the data available between the time period stat
ed
  # news <- getPage("ConnachtTribune", token, n = 10000,
  #                 since='2017/08/22', until='2018/03/22')
  # news1 <- getPage("galwaytoday1", token, n = 10000,
  #                  since='2017/08/22', until='2018/03/22')


#Merging the two news dataframes into one all_news dataframe
  # all_news <- rbind(news, news1)
  # write.csv(all_news, "All_news.csv")
## Since token is private, the facebook data extracted is stored in csv and loaded here
all_news <- read.csv("C:/Users/Suchandra/Documents/All_news.csv", stringsAsFactors = F)
```

Cleaning the posts extracted

The posts extracted from facebook have many missing messages field, different cases in writing, smileys, emojis and special charcaters which are read in R as <??>. These characters need to be removed before any further processing can be done on them.Also, words will be stemmed so as to prevent same words with singular and plural form being counted separately. Here, we have only considered all english words in language option and punctuations have been removed.

```
#Suchandra Mandal
# Removing all empty and NA entries in message field of the dataframe of all_news.
g <- all_news[!is.na(all_news$message),]

# Making a dataframe of the columns we are interested in from the previous datafarme
clean_data <- g[,c('message','likes_count', 'comments_count', 'shares_count')]

# Removing unwanted ascii characters from messages before processing
clean_data$message <- sapply(clean_data$message,function(row) iconv(row, "latin1", "ASCII", s
ub=""))

# Converting all words in message to lower case
clean_data$message <- tolower(clean_data$message)
```

2. Finding out Relevant keywords from the posts to define policies

We have some 1227 posts in total after the cleaning process. To extract the relevant keywords in order to form policies out of it, we did not want to select the words manually. Instead, we used LDA(Latent Dirichlet allocation) model to classify our posts into different topics and thereby, find relevant keywords.

For LDA model, we have used the library RtextTools. LDA is useful for finding topics and segregating the terms it comes across in unsupervised manner, that is, no previous training is required. The LDA needs input of posts in form of a document term matrix. We have used term frequency weighting scheme here to create my matrix

of terms. We have kept 40 as the number of various topics We want to look at. Usually, to determine K perfectly, people use supervised learning from previosuly classified topics data which in this case, we do not have.

Also, in order not to have any 0 entries in our matrix since that will get an error thrown by LDA model, we have calculated sum of the rows of matrix and if any row is 0, that is removed before the matrix is input the LDA model.

```
#Suchandra Mandal
# Remove overly common terms

# We want to now remove overly common terms such as galway, etc which are present
# in atleast more than 60% of the posts.
remove_terms <- function (data, threshold)
{
  stopifnot(inherits(data, c("DocumentTermMatrix", "TermDocumentMatrix")),
            is.numeric(threshold), threshold > 0, threshold < 1)
  m <- if (inherits(data, "DocumentTermMatrix"))
    t(data)
  else data
  t <- table(m$i) < m$ncol * (threshold)
  index <- as.numeric(names(t[t]))
  if (inherits(data, "DocumentTermMatrix"))
    data[, index]
  else data[index, ]
}


# Creating the document term frequency matrix by excluding numbers, stopwords, punctuations e
tc
matrix <- create_matrix( as.vector(clean_data$message),
                         language="english",
                         removeNumbers=TRUE,
                         toLower = T,
                         removePunctuation = T,
                         removeStopwords = T,
                         stemWords=TRUE,
                         weighting=weightTf)

matrix <- remove_terms(matrix,.6)
# K is the number of topics I want LDA model to extract from the posts
k <- 40

#sum by each row of the table
raw.sum=apply(matrix,1,FUN=sum)
# Exclude all 0 rows
matrix = matrix[raw.sum!=0,]

# fitting our LDA model with expected number of topics
lda <- LDA(matrix,method = "Gibbs", k)

# Groups the posts into different topics
get_topics <- topicmodels::topics(lda, 10)

## In this case I am returning the top 20 terms from each topic.
get_terms <- as.data.frame(topicmodels::terms(lda, 20), stringsAsFactors = FALSE)

# extracting the top ten topics. I chose 10 topics since more than that, the words returned a
re not much useful or give any context
f <- get_terms[1:10]

# Since some terms repeat across topics, taking a unique of the terms list
unique_terms <- unique(unlist(f,use.names = F))
```

Finding counts of the various terms extracted via LDA

Next, we wanted to look at the counts of the terms extracted and remove overly common terms. The overly common terms have to be removed since even though, they are not stopwords, the semantic context they provide is very little. Words such as Galway, will, city etc are repeated in most posts. Thus, a function was written next to remove overly common terms and a threshold was set to remove them. For the current posts, a threshold of 0.6 was kept. This means if a word is repeated in more than 60% of posts, it is not unqiue to some events and is overly common.

After this, we decile the terms and group the terms according to their counts in 10 bins. This is done in order to select the keywords from the different bins

```
#Ashfaque Azad
# creating a dataframe which stores the unique terms and a column count which will store the
 counts of each term in posts
counts <- data.frame(unique_terms,count = 0, stringsAsFactors = F)

# counting the occurence of terms in posts
for (i in 1:nrow(counts)){
  counts$count[i] <- sum(grepl(counts$unique_terms[i], clean_data$message))
}

# Removing count with 0 since only special charcters will be prsent as this is a
# subset of all messages so there must be 1 occurence
counts <- counts %>% filter(count>0) %>% arrange(count)

#print(counts)




# Putting the terms into different bins based on count
counts$bins <- ntile(counts$count,n = 10)

# Randomly selecting terms from each of the bins.
choose <- counts %>% group_by(bins) %>% sample_n(2)
#print(choose)
```

Picking interesting keywords from bins. Bin 10 has least intersting keywords since the keywords which occur in about 59% of the posts will not have much of news relevance. In contrast, bin 1 to 7 have most interesting keywords

The final set of relevant keywords chosen are news related keywords from the count and choose dataframes such as ophelia, storm, crimes, accomodation etc.

In our data, we have likes, shares and comments counts for each of the posts. A field popularity is created with the sum of the 3 fields in order for further processing to form a dataframe which will contain all the aggregates

```
#Suchandra Mandal
### set of most relevant keywords as selected from choose and count dataframe
words <- c('kayaking','investigating', 'crimes', 'damage', 'missing',
           'crowned', 'garda', 'support', 'football', 'nationwide',
           'accomodation', 'defence', 'ophelia', 'christmas', 'emergency',
           'festival', 'storm')

clean_data$Popularity <- clean_data$likes_count + clean_data$comments_count +
                         clean_data$shares_count
```

3. Filtering Relevant posts

According to the relevant keywords which have been stored in vector ‘words’, the posts having the keywords are filtered with the likes, comments, shares and popularity data stored in a new dataframe called posts_store.

```
#Suchandra Mandal
N <- 236
posts_store <- data.frame(keyword = NA, Posts = NA,
                          Popularity = integer(N),
                          likes = integer(N), comments = integer(N),
                          shares = integer(N))

# storing the relevant posts containing the keywords in a dataframe along with comments, like
s, shares and popularity count
k <- 1
  for (j in words){
    for (i in clean_data$message){
      if (sum(grepl(j, i)) == 1){
        data <- clean_data[clean_data$message == i,]
        posts_store$keyword[k] <- j
        posts_store$Posts[k] <- i
        posts_store$Popularity[k] <- data$Popularity
        posts_store$likes[k] <- data$likes_count
        posts_store$comments[k] <- data$comments_count
        posts_store$shares[k] <- data$shares_count
        k = k+1

    }
  }
}
```

4. Aggregating the data of the relevant posts

using the relevant posts dataset, we now obtain the count of posts with the keywords selected and then inner join with the posts_store dataset to keep count of posts along with relevant posts. Since keywords will repeat for may posts, counts will be repeated too. While aggregating, we need to be careful of that.

```
#Ashfaque Azad
# Summing up the posts with the keywords
keywords_count <- posts_store %>% group_by(keyword) %>% summarise(Posts_Count = n())

# Inner joining the keywords count with posts, to get counts in our dataset
posts_store <- inner_join(posts_store, keywords_count, by = "keyword")

# For each of the keywords, arranging the rows of dataframe in descending order of the Popula
rity of the keyword.
posts_store <- posts_store %>% group_by(keyword) %>% arrange(keyword,desc(Popularity))

# Writing out this aggregated dataset as a csv
  # write.csv(posts_store,"posts_store.csv")


# aggregate popularity of all policy keywords
best_policy <- posts_store %>% group_by(keyword) %>%
  summarise(Sum_popularity = sum(Popularity)) %>% arrange(desc(Sum_popularity))
```

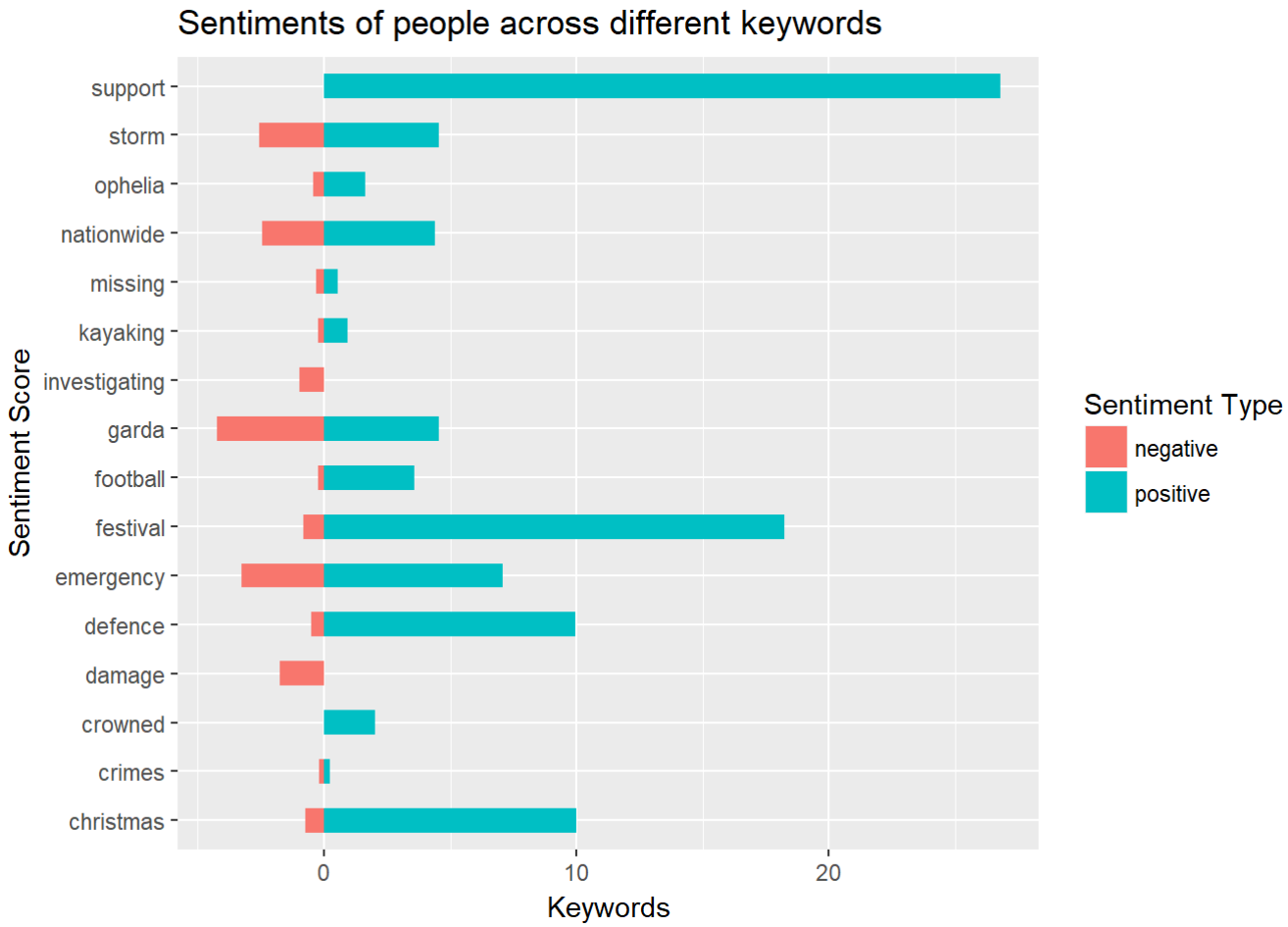
5. Visualisation of the Results:

The visualisations we have worked on are based on: a. Sentiment Analysis b. Finding most popular keywords c. A word cloud based on trending keywords d. Showing the most popular posts for each keyword as a table e. Comparison word cloud to show correlated words with each of the keywords chosen

Let’s start with sentiment analysis. Here, we wanted to take a look at what are people’s sentiments mostly at an average for the posts posted on facebook connacht tribune and galway today. To do sentiment analysis, we have used the sentimentr package and found the sentiments of people for the selected posts related to the keywords shortlisted before.

a. Sentiment Analysis and Visualisation

```
#Suchandra Mandal
# Using sentiment function to score the posts
s <- sentiment(posts_store$Posts)

# Since each post has several lines and sentiments are given at a line basis in the previous
 line, I have summarised the sentiments at an element_id level which is basically at each pos
t level
s <- s %>% group_by(element_id) %>% summarise(sentiment = sum(sentiment))

# Storing the sentiments as a separate column in posts_store
posts_store$Sentiment <- s$sentiment


# For the graph, we first need to prepare the data

# Creating a flag called sentiment_type to classify sentiments as negative or positive
posts_store$sentiment_type <- ifelse(posts_store$Sentiment < 0,
                                     "negative", "positive")

# sorting the rows of posts_store in order of sentiment value in ascending order
posts_store <- posts_store[order(posts_store$Sentiment), ]

# convert to factor to retain sorted order in plot.
#posts_store$Posts <- factor(posts_store$Posts, levels = posts_store$Posts)

# Using Diverging Barcharts to visualise sentiments of people
ggplot(data = posts_store) +
  geom_bar(aes(x= posts_store$keyword,
               y = posts_store$Sentiment,
              fill=posts_store$sentiment_type),
           stat = 'identity', width=.5) + coord_flip() +
  ggtitle("Sentiments of people across different keywords") +
  xlab("Sentiment Score") + ylab("Keywords") +
  scale_fill_discrete(name = "Sentiment Type")
```

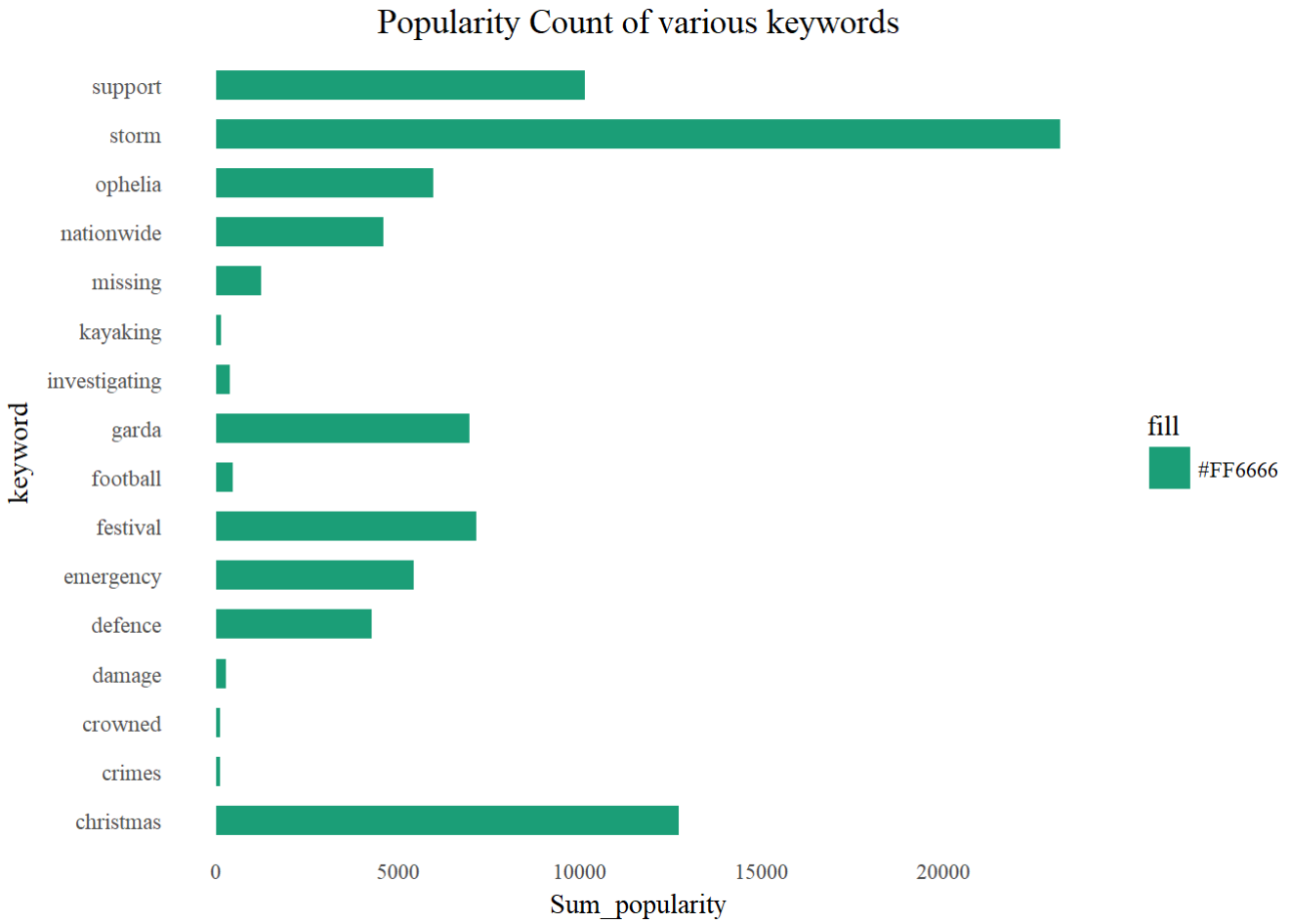


Observation: From above graph, we can see that people post mostly positive sentiments and positive sentiments far outweight those of negative

b. Finding most popular keywords and visualise them

Based on popularity of the keywords chosen the next bar graph is generated. Here, we use popularity index which is the sum of shares, comments and likes count for all posts under the particular keyword.

```
#Suchandra Mandal
options(scipen = 999)  # turns of scientific notations like 1e+40



# getting the best keywords by inner joining the most popular posts dataframe and best policy
 in order to have the most popular keywords along with their popularity and sentiment values

# The top posts in terms of popularity with respect to a keyword is stored here
most_popular_posts <- posts_store %>% group_by(keyword) %>% top_n(1,Popularity)

best <- inner_join(best_policy, most_popular_posts, by = "keyword") %>%
  arrange(desc(Sum_popularity))

# Plot
ggplot(best, aes(x = keyword, y = Sum_popularity, fill = "#FF6666")) +
  geom_bar(stat = "identity", width = .6) +
  coord_flip() +  # Flip axes
  labels(title="Popularity Count of various keywords") +
  theme_tufte() +  # Tufte theme from ggfortify
  theme(plot.title = element_text(hjust = .5),
        axis.ticks = element_blank()) +   # Centre plot title
  scale_fill_brewer(palette = "Dark2")  # Color palette
```



As expected, words such as storm which has all terms of Ophelia, Eleanor and recent rainstorms are most common followed by support and festival which could be due to the St. Patrick's day which occurred recently.

c. A word cloud based on trending keywords

Next, based on the posts of relevant posts subset before, we made a corpus object of the posts of facebook and removed all unnecessary words such as stopwords, numbers etc and then plotted a wordmap with 2 options. The first wordmap is based on the maximum number of words to show while the second limits the overly common words by putting a threshold on maximum frequency of the word in posts.

```
#Ashfaque Azad
library(wordcloud)

# converting the posts dataframe to a corpus object
corpusdata <- Corpus(VectorSource( posts_store$Posts))

# Removing stopwords, punctuations, numbers and a few words from the corpus
corpus <- tm_map(corpusdata, removeWords, stopwords("english"))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)

corpus <- tm_map(corpus, removeWords, c(letters, 'galway','will','day',
                                        'today', 'city','thursday'))


# Creating a term document matrix of the corpus
dtm <-TermDocumentMatrix(corpus,
    control = list(weighting = function(x) weightTfIdf(x, normalize = TRUE)))


# looking out the correlated terms for flood
findAssocs(dtm, terms = "flood", corlimit = 0.5)
```

```
## $flood
##   defence    called  disaster permanent structure  tuesdays      wake
##      0.58      0.55      0.55      0.55      0.55      0.55      0.55
```

```
# creating a word cloud with maximum 200 words shown
wordcloud(words = corpus,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```


```
# We can also visualise the word cloud based on frequency of words
wordcloud(words=corpus,max.freq = 200, scale=c(5,0.75), random.order=FALSE, rot.per=0.35, us
e.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```

d. Showing the most popular posts for each keyword as a table

This dataframe was already aggregated before. Just printing out the table here. The most popular post under each keyword was filtered based on popularity index and displayed as a table

```
#Ashfaque Azad
library(kableExtra)
knitr::kable(most_popular_posts[, 1:2], "html") %>% kableExtra::kable_styling("striped")
```

keyword	Posts
nationwide	<u+26a0><u+fe0f> status red snow/ice warning <u+26a0><u+fe0f> met ireann has just upgraded the red warning nationwide <u+26a0><u+fe0f> all schools, third level institutions and colleges of further education will be closed tomorrow and friday, as instructed by the minister for education due to the status red weather warning. further heavy snow showers will bring accumulations of significant levels with all areas at risk. blizzard conditions will develop from the south thursday afternoon and evening as heavy snow and strong easterly winds bring snow drifts northwards over the country. eastern and southern coastal counties will be worst affected. the warning is in place from now until 3pm friday.
defence	tides breach coastal defences in salthill. video by david blevins, sky.

keyword	Posts
investigating	garda are investigating an armed robbery at derrydonnell filling station near athenry.
damage	almost 5,000 worth of damage was caused to an inis mr pub during a burglary on saturday morning.
ophelia	#ophelia update from met ireann
storm	dramatic footage of #stormeleanor flooding from salthill, fr griffin road and the spanish arch in galway. (video: joe oshaughnessy)
festival	galway airport is set to play host the two-day fever pitch music festival this summer, with chart-toppers ub40 and rudimental already signed up as headline acts.
christmas	the big wheel at the christmas market is stuck, with people on board. fire and rescue galway and garda are at the scene. (video: enda cunningham) http://connachttribune.ie/fire-services-work-rescue-passengers-stuck-big-wheel-440/ (http://connachttribune.ie/fire-services-work-rescue-passengers-stuck-big-wheel-440/)
garda	the big wheel at the christmas market is stuck, with people on board. fire and rescue galway and garda are at the scene. (video: enda cunningham) http://connachttribune.ie/fire-services-work-rescue-passengers-stuck-big-wheel-440/ (http://connachttribune.ie/fire-services-work-rescue-passengers-stuck-big-wheel-440/)
emergency	unsung heroes … the radiation therapy team at uhg braved the elements to keep the show on the road. many other hospital staff and emergency workers have also worked through challenging conditions over the past two days. their efforts shouldnt be forgotten. photo: ann-marie ohara)
crimes	there are now 108 community text alert schemes spread across co. galway and according to a senior garda they are playing key roles in helping to prevent a range of crimes being committed and especially so in more rural areas.
crowned	congrats to calasanctius college oranmores u14 boys who were crowned all-ireland schools a basketball champions following a 51-27 win over st pats of cavan in the national basketball arena in tallaght this afternoon.
missing	update from garda: francis has been located safe and well. <u+26a0><u+fe0f> missing person<u+26a0><u+fe0f> please share garda are seeking the public’s help in tracing francis kavanagh, 61, who is missing from his home at sruthan mhuirlinne, ballybane in galway since last thursday (january 18). he is described as 6’ 2’’, of medium build with short grey hair and blue eyes. it is not known what he was wearing when he went missing. francis is believed to be using a black bmw 07 so 2426. anyone with information is asked to contact galway garda station on <u+202a>091 538000<u+202c> or any garda station.

keyword	Posts
support	<ed><u+00a0><u+00bc><ed><u+00b7><u+00b6><ed><u+00a0><u+00bc><ed><u+00b7><u+00a6><ed><u+00a0><u+00bc><ed><u+00b7><u+00b6><ed><u+00a0><u+00bc><ed><u+00b7><u+00a6> homecoming <ed><u+00a0><u+00bc><ed><u+00b7><u+00b6><ed><u+00a0><u+00bc><ed><u+00b7><u+00a6><ed><u+00a0><u+00bc><ed><u+00b7><u+00b6><ed><u+00a0><u+00bc><ed><u+00b7><u+00a6> the senior hurlers' dance moves on stage made it all worthwhile for the 30,000+ supporters who came to pearse stadium last night to join in the all-ireland celebrations!! well done to everyone who came out to support the two teams. (video: joe o'shaughnessy)
football	another great win for galway over mayo this afternoon in the allianz football league. it ended galway 1 13, mayo 0 11 at pearse stadium. congratulations lads, thats three wins out of three in the league so far for galway.
kayaking	help to bring the remains of david and alex home. on saturday, 20th january, nui galway student david higgins, of tralee, and alex mcgourty of calry, co sligo, lost their lives whilst kayaking the abianco river, in the macas region, ecuador. kayaking was david and alex's passion, and they spent their lives travelling and dreaming of challenges and experiences. meeting new people and spreading a zest for life with everyone who was lucky to meet them. they were vibrant, kind souls who lived life to its full potential, and words cannot describe what they meant to their family and friends. now, it's time to bring them home one last time. to help david's and alex's family do this, their friends have set up a gofundme page to gather funds to support their journey. any help is appreciated, no matter how small.

e. Comparison word cloud to show correlated words with each of the keywords chosen

Finally, we wanted to take a look at the associated words with each of the 4 keywords we have chosen to display. A correlation threshold was placed to extract the most meaningful terms associated with each of the keywords. As we can see, with crimes keyword we have computer, track, spread etc .

```
#Ashfaque Azad
# We selected the 4 keywords for comparison which looked most promising

#Defence
defence <-findAssocs(dtm, terms = "defence", corlimit = 0.3)[[1]]
defence
```

```
##         civil         flood       standby         boxer       funding
##          0.64          0.58          0.46          0.45          0.45
##         kevin         moran          fire      minister          tide
##          0.45          0.45          0.42          0.40          0.39
##       million         works           opw       service        called
##          0.39          0.39          0.38          0.35          0.35
##      disaster     permanent     structure      tuesdays          wake
##          0.35          0.35          0.35          0.35          0.35
##     appointed     blueprint comprehensive   consultants          draw
##          0.33          0.33          0.33          0.33          0.33
##        months        scheme     scheduled      tomorrow        braced
##          0.33          0.33          0.32          0.32          0.31
##     elaborate      erecting    inflatable        rising        safely
##          0.31          0.31          0.31          0.31          0.31
```

```
defence <- cbind(read.table(text = names(defence), stringsAsFactors = FALSE),defence)

defence$V1[1:12]
```

```
##  [1] "civil"    "flood"    "standby"  "boxer"    "funding"  "kevin"
##  [7] "moran"    "fire"     "minister" "tide"     "million"  "works"
```

```
defencev1 <- Corpus(VectorSource(defence$V1))

#wordcloud(words = defencev1, random.order=FALSE, scale=c(1,0.75), colors = brewer.pal(4, "Da
rk2"))

#crimes
crimes<- findAssocs(dtm, terms = "crimes", corlimit = 0.4)[[1]]
crimes
```

```
##   computer  criminals        log      pulse   stations      track
##       0.87       0.87       0.87       0.87       0.87       0.87
##     access       used     system     spread especially    prevent
##       0.83       0.75       0.70       0.49       0.49       0.49
##      roles       text      rural  committed    helping      alert
##       0.49       0.49       0.48       0.48       0.45       0.44
##    schemes
##       0.42
```

```
crimes <- cbind(read.table(text = names(crimes), stringsAsFactors = FALSE), crimes)

crimes$V1
```

```
##  [1] "computer"   "criminals"  "log"        "pulse"      "stations"
##  [6] "track"      "access"     "used"       "system"     "spread"
## [11] "especially" "prevent"    "roles"      "text"       "rural"
## [16] "committed"  "helping"    "alert"      "schemes"
```

```
crimesv1 <- Corpus(VectorSource(crimes$V1))

#wordcloud(words = crimesv1, random.order=FALSE, scale=c(1,0.75), colors = brewer.pal(4, "Dar
k2"))
#storm
storm<-findAssocs(dtm, terms = "storm", corlimit = 0.30)[[1]]
storm
```

```
##       eleanor   devastation     executive          says unprecedented
##          0.60          0.50          0.50          0.50          0.50
##       brewing      stunning          band         loved         chief
##          0.49          0.49          0.48          0.43          0.40
##          much         warns      restored          play        called
##          0.39          0.36          0.36          0.32          0.32
##      disaster     permanent     structure      tuesdays          wake
##          0.32          0.32          0.32          0.32          0.32
##     legendary         homes
##          0.32          0.31
```

```
storm <- cbind(read.table(text = names(storm), stringsAsFactors = FALSE), storm)

storm$V1
```

```
##  [1] "eleanor"       "devastation"   "executive"     "says"
##  [5] "unprecedented" "brewing"       "stunning"      "band"
##  [9] "loved"         "chief"         "much"          "warns"
## [13] "restored"      "play"          "called"        "disaster"
## [17] "permanent"     "structure"     "tuesdays"      "wake"
## [21] "legendary"     "homes"
```

```
stormv1 <- Corpus(VectorSource(storm$V1))

#wordcloud(words = stormv1, random.order=FALSE, scale=c(1,0.75), colors = brewer.pal(4, "Dark
2"))
#ophelia
ophelia <- findAssocs(dtm, terms = "ophelia", corlimit = 0.30)[[1]]
ophelia
```

```
##     salthill       update          met        video       ireann
##         0.50         0.45         0.43         0.43         0.39
##          joe     closures        worst  exhurricane    aftermath
##         0.34         0.31         0.31         0.31         0.31
##      carroll     prepared        brunt    finalised preparations
##         0.31         0.31         0.31         0.31         0.31
##        later
##         0.30
```

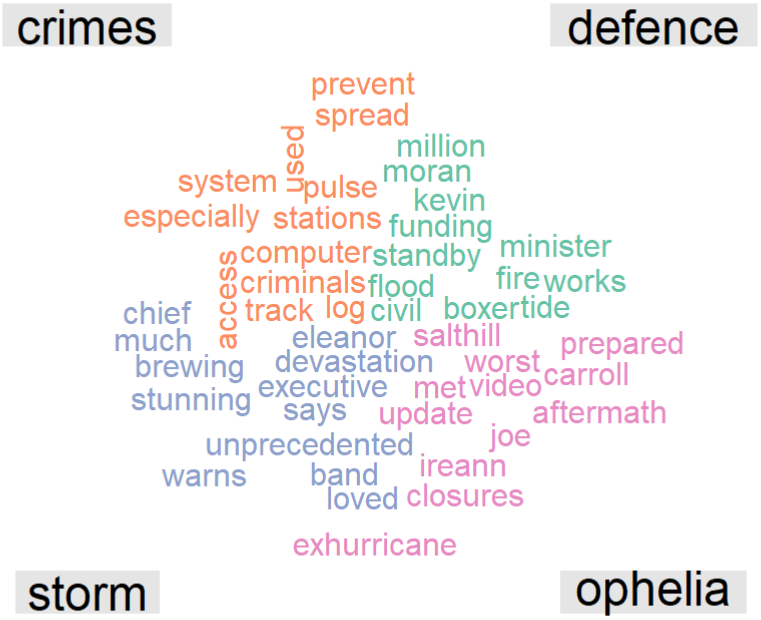
```
ophelia <- cbind(read.table(text = names(ophelia), stringsAsFactors = FALSE), ophelia)

ophelia$V1
```

```
##  [1] "salthill"     "update"       "met"          "video"
##  [5] "ireann"       "joe"          "closures"     "worst"
##  [9] "exhurricane"  "aftermath"    "carroll"      "prepared"
## [13] "brunt"        "finalised"    "preparations" "later"
```

```
opheliav1 <- Corpus(VectorSource(ophelia$V1))

#wordcloud(words = christmasv1, random.order=FALSE, scale=c(1,0.75), colors = brewer.pal(4,
 "Dark2"))

classic4 <- c(defence$V1[1:12],crimes$V1[1:12],storm$V1[1:12], ophelia$V1[1:12])
classic4
```

```
##  [1] "civil"         "flood"         "standby"       "boxer"
##  [5] "funding"       "kevin"         "moran"         "fire"
##  [9] "minister"      "tide"          "million"       "works"
## [13] "computer"      "criminals"     "log"           "pulse"
## [17] "stations"      "track"         "access"        "used"
## [21] "system"        "spread"        "especially"    "prevent"
## [25] "eleanor"       "devastation"   "executive"     "says"
## [29] "unprecedented" "brewing"       "stunning"      "band"
## [33] "loved"         "chief"         "much"          "warns"
## [37] "salthill"      "update"        "met"           "video"
## [41] "ireann"        "joe"           "closures"      "worst"
## [45] "exhurricane"   "aftermath"     "carroll"       "prepared"
```

```
classic4 <- Corpus(VectorSource(classic4))

classic4.tdm <- TermDocumentMatrix(classic4)
library(magrittr)
facebook.tdm.matrix <- classic4.tdm %>% as.matrix()
f <- sort(rowSums(facebook.tdm.matrix),decreasing=TRUE)
d <- data.frame(word = names(f),freq=f)


defence <- rowSums(facebook.tdm.matrix[,1:12])
crimes <- rowSums(facebook.tdm.matrix[,13:24])
storm <- rowSums(facebook.tdm.matrix[,25:36])
ophelia <-rowSums(facebook.tdm.matrix[,37:48])

# create a data frame
classic.summary <- data.frame(defence,crimes,storm, ophelia)
# scale the columns so that larger corpii don't dominate
classic.summary.scaled <- scale(classic.summary[,1:4],center=FALSE, scale=colSums(classic.sum
mary[,1:4]))

par(mfrow=c(1,1))
comparison.cloud(classic.summary.scaled,scale=c(1,0.75), random.order=FALSE, colors= c('#66c2
a5','#fc8d62','#8da0cb','#e78ac3','#a6d854'),
                 title.size=1.5)
```


crimes
defence
prevent
spread
used
million
moran
system
pulse
kevin
especially
stations
funding
computer
standby
minister
access
criminals
flood
fire
works
chief
track
log
civil
boxertide
much
eleanor
salthill
prepared
brewing
devastation
worst
carroll
executive
met
video
stunning
says
update
aftermath
joe
unprecedented
ireann
warns
band
loved
closures
exhurricane
storm
ophelia