Data Science Week 10: Topic Modelling Mozdeh Tweets with R

# Objectives

* Learn how to use R for topic modelling of tweets collected by Mozdeh.
* Learn how to interpret vague topics produced by R topic models of tweets.

# R script

The statistical and data science program R can be used for topic modelling using one of its specialist packages. R also includes natural language programming packages that can perform stemming, stop word removal and other tasks on sets of texts. It can also split large documents into bags of words.

In this module we will use a pre-written R script to carry out the tasks. You won’t need to write your own R script in this module but it is important to understand how the R script works. In data science, it is common to recycle other people’s code, at least to start with. The first part of the script lists the R packages used.

library(tm) #text mining library

library(SnowballC) #for the word stemming

library(tictoc) #timing library

library(topicmodels) #load topic models library

library(tidytext) #Formatting library

library(ggplot2) #Graphing library

library(dplyr) #Formatting library

You will need to edit the next part of the script to point to the folder in which Mozdeh has stored the news tweets. You will need to find it on your computer using Windows Explorer and edit the script, using double slashes for folder names. This is the main part of the script that you need to edit.

setwd("E:\\rss\_data\\Brexit\\raw data")

The script should already have the filename of the tweets entered, plus the name of the column in the file containing the tweets.

filteredTweetsFile <- "TwitterSearches\_Tweets\_AllFiltered.txt"

TextColName <- "Tweet.Title."

The next part of the script extracts the tweets and saves them to a new file.

Tweets.df <- read.csv(filteredTweetsFile, sep = "\t", header = TRUE, quote = "", skipNul=TRUE)

Tweets.df <- Tweets.df[TextColName] #Tweets.df["Tweet.Title."]

filteredTweetsFileTweetsOnly <- paste(filteredTweetsFile, "TweetsOnly.txt", sep="")

write.table(Tweets.df, filteredTweetsFileTweetsOnly, row.names = FALSE, quote=FALSE)

The tweets are then read from the new file.

con <- file(filteredTweetsFileTweetsOnly, "rt")

readLines(con, n=1) #read and throw away the first line - the heading

tweets = readLines(con) #Read all the tweets

close(con)

Tweets contain lots of strange characters and URLs, so the bizarre but powerful function gsub is used to get rid of them. You don’t need to understand how it works.

tweets = gsub("(RT|via)((?:\\b\\W\*@\\w+)+)","",tweets)

tweets = gsub("http[^[:blank:]]+", "", tweets) #Removes hyperlinks from tweets

tweets = gsub("@\\s+", "", tweets) #removes @usernames from tweets

tweets = gsub("&\\w+", "", tweets) #removes &amp; and other escape characters

tweets = gsub("^\\s+|\\s+$", "", tweets) #trim whitespace from start

tweets = gsub('\\d+', '', tweets) #Removes numbers from tweets

tweets = gsub("[[:punct:]]", " ", tweets) #Removes punctuation from tweets

tweets = gsub('[^\x20-\x7E]', '', tweets) #Removes non-Ascii from tweets

Next, the tweets are converted into a document collection called a corpus, and the text is standardised more. You may guess what most of the commands do from their names.

corpus = Corpus(VectorSource(tweets)) #Corpus object from R package tm

corpus = tm\_map(corpus,removePunctuation)

corpus = tm\_map(corpus,tolower) #Converts tweets to lowercase

corpus = tm\_map(corpus,removeWords,stopwords("english"))

corpus = tm\_map(corpus,stripWhitespace)

corpus = tm\_map(corpus,stemDocument)

The document/word frequency matrix to be used for topic modelling is now created. This is like the matrices that we created at the end of last week.

dtm <- DocumentTermMatrix(corpus)

sparseValue <- .99 #To remove rare words (occurring in less than 1% of tweets)

dtm <-removeSparseTerms(dtm, sparse=sparseValue)

Set the number of topics to be extracted, k. The topic modelling algorithm used by R does not work this out itself, so you have to guess this one. The strange parameters in the LDA function are not relevant to this module so you can ignore them. After this, the topic modelling command is run, which may take a few minutes.

k <- 6

ldaOut <- LDA(dtm,k, method="Gibbs", control=list(nstart=nstart, seed=seed, best=best, burnin=burnin, iter=iter, thin=thin))

The results are then saved into different files. For the rest of this session we will focus on understanding the information in these files and reported by R.

# Topic modelling output from R

The topic modelling output includes a list of the topics extracted. The topics are not named by R but the words with the highest probabilities for the topic are listed and the next task is to use the words, and searches of the tweets matching the words, to decide what to call the topics. Many topics do not have an obvious name or connection, so this stage is really really tricky.

For the six topics from 7 August 2019, the clearest topic seems to be 5, which is about Donald Trump’s comments on a mass shooting of people in El Paso (hence the terms Trump, people, Paso, shoot, mass). Note that Porter stemming is used, so the term “shoot” probably mainly refers to the term “shooting”. The other terms have a less clear association, but do not point to other topics, so we are safe to label topic 5, “Trump’s reaction to the El Paso shooting”. The other topics are much harder to detect.

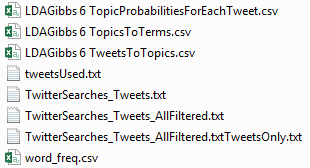
Searches of Mozdeh for the keywords, year, say, and time, suggest that many of the tweets are crime reports, giving the age of the offender or victim, and sometimes the length of time the offender has previously served in prison. Topic 2 could therefore be called, “Crime”.

The other topics do not have a clear description or explanation. Topic 4, with the main keyword “will” seems to relate to news reports of things that are about to happen, so it is a future orientation pattern rather than a real topic. It could be called “Future orientation”. I am not sure that the other topics can be given useful labels.

The reason why most of the topics can’t be labelled are that (a) we don’t have much data, and (b) the data we have covers so many different topics that topic modelling can’t detect them all.

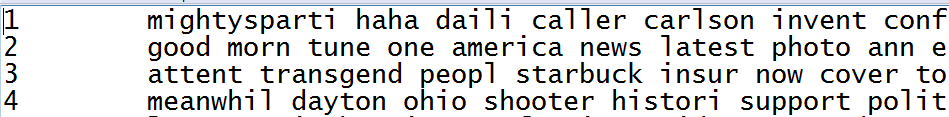
E:\rss_data\Twitter news\raw data\6 topics.tiff

The other outputs produced by R will be saved in the raw\_data folder alongside the original tweets, which should look as follows:

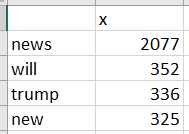


The main files are as follows contain:

* **tweetsUsed.txt**: Lists all tweets included in the analysis, one per line. Double click it to load it in Windows Notepad. The tweets look like gibberish – why do you think this is?



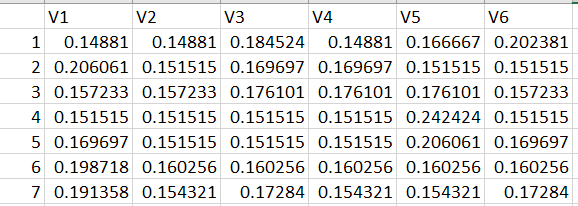
* **word\_freq.csv**: Lists all words extracted from the tweets, after stemming, and their overall frequency. Double click it to open it in Excel, In the extract below, the top four words are news (actually the hashtag #news), will, Trump, and new (probably the word news, stemmed).



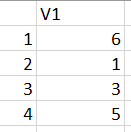
* **LDAGibbs 6 TopicsToTerms.csv**: Gives the same information as the topic graphs, but gives a longer list of terms.



* **LDAGibbs 6 TopicProbabilitiesForEachTweet.csv**: Lists the probabilities that each tweet (row) contains the topic (column). For example, tweet 1 is most likely to contain topic 6. Recall that topic 5 was the clearest one, and tweet 4 above is clearly about that. This is reflected in the V5 column probability being the highest column value for Tweet 4.



* LDAGibbs 6 TweetsToTopics.csv: Lists the main topics for each tweet. In the example below tweet 4 is assigned to topic 5, as expected.



# Workshop and tutorial

If you have not already done so, following the week 1 instructions, use Mozdeh to collect at least 5 minutes of tweets for the keyword search *news*. We will use this set of tweets (hopefully at least 2000, to give plenty of texts to analyse) to practice topic modelling with R. While Mozdeh is running, there is time to setup R as follows.

1. Start R and install the following packages: tm; SnowballC; tictoc; topicmodels; tidytext; ggplot2; dplyr. For each one, select *Tools > Install packages* and then enter the name of the package (case sensitive) and download it.
2. Download the R script TopicModelTwitter.R from the module page in Canvas and load it into R with *File > New file > R Script*, selecting this script file.
3. Near top of TopicModelTwitter.R, edit the line setwd("E:\\rss\_data\\Brexit\\raw data") to point to the raw data folder of your news project. You will need to find this folder. It will be in a subfolder of the folder from where you started Mozdeh. Remember to use double slashes for the filename. Do NOT run the command yet.
4. Switch to Mozdeh, click the stop button in Mozdeh, and wait for it to index the file and click yes or OK to the questions asked until it gets to the main search screen.
5. Switch back to R and run your modified version of the command setwd("E:\\rss\_data\\Brexit\\raw data") by clicking on it and clicking the Run button, and check that it does not return an error message. If it does, fix the path until it works.
6. Select the whole program and run it, checking for any errors. If you find errors, fixt them and try again. When the program is running, it pauses for up to 5 minutes at the line starting ldaOut <- because this is when it is solving the big matrix factorisation problem of topic modelling.

The output of Mozdeh is a set of graphs in the bottom right hand corner and a set of files containing details of the topic models. You now must interpret the results and then give the topics a meaningful name, when relevant. Some of the topics might be tweets about current news stories, whereas others might be meaningless. To interpret the topics, look at the words and use common sense to match them to current news stories. Also to help with the matching, search the terms in Mozdeh and read some of the matching tweets.

* Task 1: Try to find a meaningful name for at least one of your topics.
* Task 2: open and investigate the files in the raw\_data folder, checking that they contain the information stated in the description above. Use the information in these files to find a tweet that matches the topic that you found in Task 1.

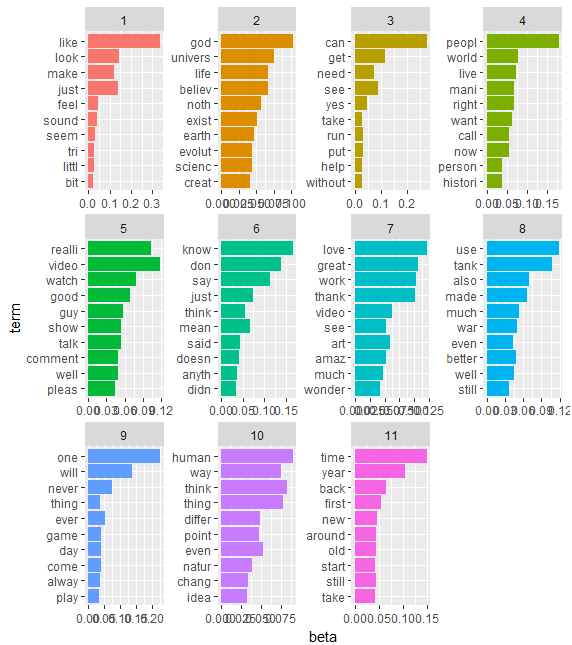
# Extra examples

In the example below from 7 August 2019, some logical topic names are as follows. Note that hashtags are removed before the topic modeling.

1. Crime: Seems to be about multiple crime news stories mentioning the age of the victim or offender.
2. Say: This does not seem to be a topic, but seems to be the “topic” of tweeting what others “say” in the news.
3. Can: This also does not seem to be a topic, but seems to be the “topic” of tweeting what “can” be done in some news situations.
4. Spam: Tweets matching these searches tended to have long lists of hashtags and look spammy.
5. Will: Tweets mentioning the future.
6. Likes: Positive reactions to news events.
7. Fake news: Fake new and the media.
8. Shootings: Donald Trump’s reaction to two mass shootings within 24 hours in the USA.
9. Reports: News stories about just-released reports.
10. Commentaries: Lots of tweets of people reporting their reactions (e.g., “I just want to say that…”)
11. Not sure
12. Fox news.

E:\rss_data\Twitter news\raw data\12 news topics.tiff

The graphs below are for comments about museum videos on YouTube. What names would you give to topics 2, 7, and 8?

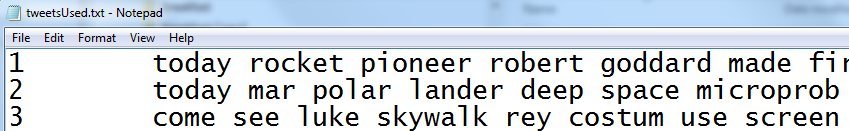


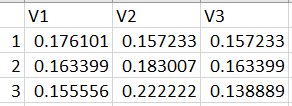
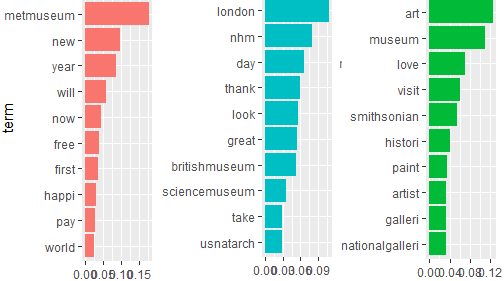
# Exam questions

1. The screenshot below was taken from Mozdeh after running topic modelling on a set of 12602 breakfast tweets collected.
   1. The word *eat* is second for topic 3. How would you interpret the longer bar for *eat* in comparison to the term *lunch* in the graph for topic 3, in terms of topic modelling algorithm probabilities?
   2. Why do you think that the word *breakfast* is at the top of all topics?
   3. Think of an appropriate description for topics 4 and 6. There is no single correct answer.
   4. Do you think that “People loving breakfast time” is a possible description for Topic 1? Explain your answer.
   5. The term *serv* is seventh for topic 2. Since *serv* is not a word, why do you think it might be in this list?

E:\rss_data\breakfast\raw data\Breakfast 6 topics.tiff

1. The screenshots below were taken from Mozdeh after running topic modelling on a large set of museum-related tweets.
   1. The tweetsUsed.txt file contains the text of the tweets processed for the topic modelling. Why do the words in the tweets seem not to make sense?
   2. Give topic names for the three topics illustrated below from this collection. To might like to know that #MetMuseum is for a New York museum and #NHM is the Natural History museum in London, and the Smithsonian American Art Museum is famous.
   3. From any or all of these three screenshots, which topics are the tweets from? Justify your answers.





# Exam Answers

1. .
   1. Eat has a higher probability to be selected when discussing topic 3.
   2. Because all the tweets contain the term breakfast as it was the search term.
   3. Breakfast food/breakfast morning; Don’t want to get out of bed for breakfast/Don’t want to cook breakfast.
   4. Yes because words related to this phrase are in the top topic terms list.
   5. A stemmed form of serving – for discussing “serving breakfast”.
2. Because they have been stemmed, converted to lower case and punctuation removed
3. Met Museum; London museums; Art museums/galleries [Smithsonian American Art Museum also acceptable]
4. First, second, second, using the highest p value in the row.