THagen\_Mod5\_Assignment2: Structural Topic Modeling

Tabitha Hagen

4/12/2023

# 1 - Load the required libraries

# 2 - Download the data

### The dataset used in this assignment contains 10,000 records extracted from twitter for trending topic of - #AvengersEndgame (<https://www.kaggle.com/datasets/kavita5/twitter-datasetavengersendgame?resource=download>).

# Read in the data  
data<-read.csv("tweets.csv")

# 3 - Select the tweet’s text and retweetCount variable (that describes the number of retweets a tweet gets):

tweets<- data %>%  
 select(X,text,retweetCount)

# 4 - Run an STM model:

## • The textProcessor function of the stm package asks us to specify the part of the dataframe where the documents we want to analyze are (tweets$text). It also requires us to name the dataset where the rest of the meta data live (metadata=tweets).

### The textProcessor function automatically removes a) punctuation; b) stop words; c) numbers, and d) stems each word. The function requires us to specify the part of the dataframe where the documents we want to analyze are (ours are called tweets$text), and it also requires us to name the dataset where the rest of the meta data live (metadata=tweets).

processed <- textProcessor(tweets$text, metadata=tweets,   
 removestopwords = TRUE,  
 removenumbers = TRUE,  
 removepunctuation = TRUE,  
 ucp = TRUE,  
 wordLengths = c(3, Inf),  
 customstopwords=c("https", "coxcwkcmww", "til", "copvchpb", "say", "six", "get", "skskskskss", "oppomobilendia", "cozhodjw", "coszfbsggg", "covlpepnxygm", "counyigww", "cozhodjw", "cogobzxkktdd", "cojkoqqobsi", "edit"),  
 onlycharacter = TRUE)

## Building corpus...   
## Converting to Lower Case...   
## Removing punctuation...   
## Removing stopwords...   
## Remove Custom Stopwords...  
## Removing numbers...   
## Stemming...   
## Creating Output...

## • The stm package also requires us to store the documents, meta data, and “vocab”—or total list of words described in the documents—in separate objects. The first line of code eliminates both extremely common terms and extremely rare terms, as is common practice in topic modeling, since such terms make word-topic assignment much more difficult.

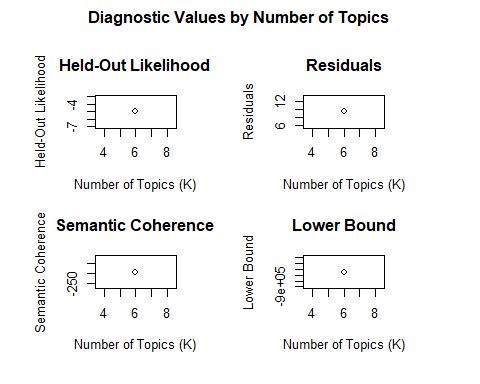
out <- prepDocuments(processed$documents, processed$vocab, processed$meta)

## Removing 3301 of 6074 terms (3301 of 128314 tokens) due to frequency   
## Removing 2 Documents with No Words   
## Your corpus now has 14998 documents, 2773 terms and 125013 tokens.

docs <- out$documents  
vocab <- out$vocab  
meta <-out$meta

## • We can find the number of k using the function search(). The model uses the “retweetCount” variable to improve topic classification (This step may take a long time to run. pick k=6 if that is the case. you can see in the diagrams below that k=6 would be the optimal number of topics.)

# changed K from K=c(5:7) to K=c(6)  
findingk <- searchK(out$documents, out$vocab, K = c(6),  
prevalence = ~ retweetCount, data = meta, verbose=FALSE)  
plot(findingk)

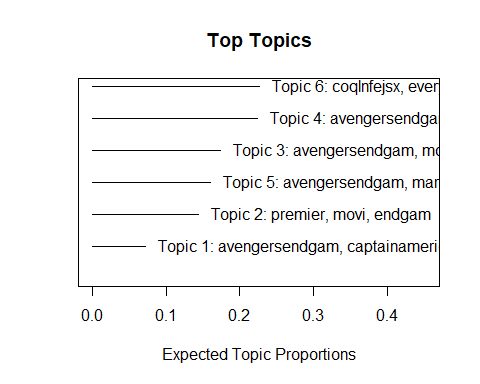


###Now, we are going to run our stm model. Before we run the model, readers should also note that the STM package also has an argument that allows one to specify the type of initialization or randomization that should be used—in this case we are using spectral initialization, which has several advantages over a random seed that are discussed in the paper linked above.

tweet\_STM <- stm(documents = out$documents, vocab = out$vocab,  
 K = 6, prevalence =~ retweetCount,  
 max.em.its = 75, data = out$meta,  
 init.type = "Spectral", seed=100, verbose = FALSE)

## • Plot the results. The stm package has a useful function that visualizes these results called plot ().

plot(tweet\_STM, type = "summary")

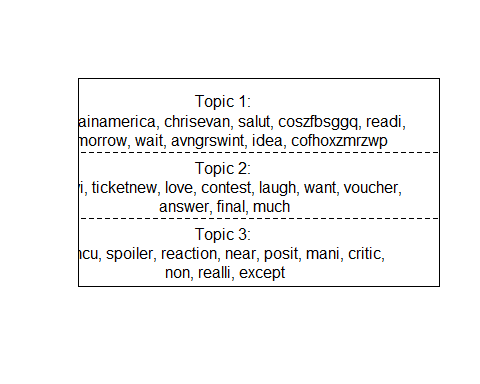


## • You can also visualize a list of the top 10 words per specific topic: labelTopics() and plot.STM().

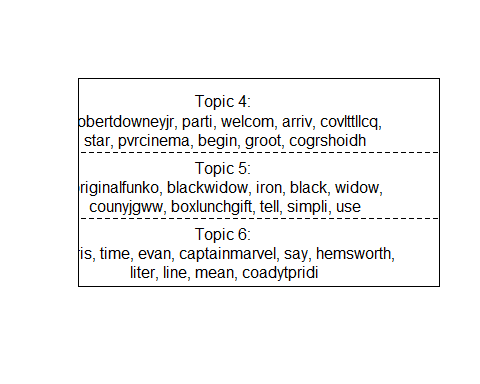
labelTopics(tweet\_STM, topics=c(1,4), n=10)# complete list of top 10 words per topics 3,4

## Topic 1 Top Words:  
## Highest Prob: avengersendgam, captainamerica, chrisevan, salut, marvel, coszfbsggq, readi, tomorrow, wait, avngrswint   
## FREX: captainamerica, chrisevan, salut, coszfbsggq, readi, tomorrow, wait, avngrswint, idea, cofhoxzmrzwp   
## Lift: ahead, alejandrasoto, allinfinitylov, babucinema, battl, bean, bestofbradga, besttraiier, bingthisway, bye   
## Score: salut, captainamerica, chrisevan, coszfbsggq, readi, cofhoxzmrzwp, idea, avngrswint, tomorrow, avengersendgam   
## Topic 4 Top Words:  
## Highest Prob: avengersendgam, man, aveng, marvel, chanc, follow, marvelstudio, ironman, robertdowneyjr, parti   
## FREX: robertdowneyjr, parti, welcom, arriv, covlttllcq, star, pvrcinema, begin, groot, cogrshoidh   
## Lift: coaagrbvb, smash, abbrevi, agent, agentsofshield, arstechnica, awesomemerg, begin, brand, briannacherri   
## Score: man, aveng, marvelstudio, arriv, welcom, follow, coqnsmdcdm, ironman, robertdowneyjr, surpris

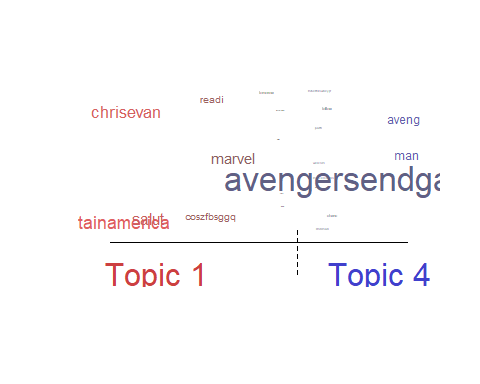
plot.STM(tweet\_STM, type="labels", topics=c(1:3), label="frex", n=10, width=55)#top 10 FREX words per topics 1-3



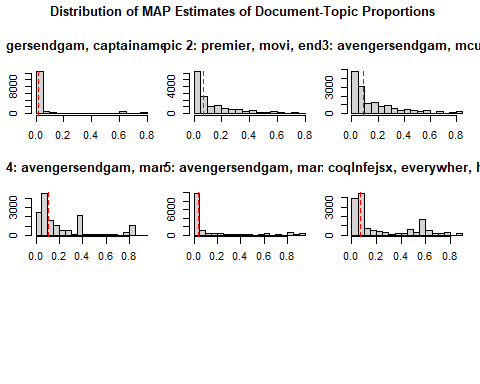
plot.STM(tweet\_STM, type="labels", topics=c(4:6), label="frex", n=10, width=55)#top 10 FREX words per topics 4-6



plot(tweet\_STM, type="perspectives", topics=c(1,4)) #differences between two topics, content covariates or combinations.



plot(tweet\_STM, type="hist") #histogram of the expected distribution of topic proportions across the documents



## • Explore the visualizations and interpret the results.

The 1st visualization is a summary which shows the top 10 words for Highest Probabiliy, FREX, LIFT and SCORE methods. The 1st topic has Captain America, tomorrow and wait as important words whiles the 4th Topic has Avenger, Ironman, Robet Downey, Jr, and welcome. The next visualization is two pages of clear charts, listing the 10 most important words according to only FLEX. Again, Captain America, Iron and Robert Downey Jr. are important. The 3rd visualization has a chart comparing the words of 2 topics with the most important words appearing larger: Captain America stands out along with Avenger. The last visualization are histograms of each topic and you can see Avengers in Topics, 3, 4, 5, and 6.

# 5 - We can take the trained topic model and, using some supplementary metadata on our documents, estimate regressions for the proportion of each document about a topic with the metadata as the predictors. For example, we can see how tweets about certain topics are more or less likely to be retweeted.

## Use the function: estimateEffect() - Your formula would be: c(Topic i)~ retweetCount - For example, we can see how tweets about topic 5 [actually 4] are related to retweet counts:

tweet\_predict\_topics<-estimateEffect(formula = c(4)~ retweetCount, stmobj =  
tweet\_STM, metadata = out$meta, uncertainty = "Global")

# 6 - Explore a summary of the results using the summary () function:

## See if there is statistical evidence of change in the tweet topic proportion according to retweet count! (For example, based on my analysis, I can see topic 5 [actually Topic 4] (that include the word Robert Downey Jr.) is likely to get significantly more retweets (Estimate: 2.387e-05 and p-value<0 \*\*\*).

summary(tweet\_predict\_topics)

##   
## Call:  
## estimateEffect(formula = c(4) ~ retweetCount, stmobj = tweet\_STM,   
## metadata = out$meta, uncertainty = "Global")  
##   
##   
## Topic 4:  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.670e-01 2.788e-03 59.89 <2e-16 \*\*\*  
## retweetCount 9.273e-06 3.085e-07 30.05 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# 7 - Write a paragraph of your findings (less than 200 words).

The Structural Topic Model (STM) allows us to use not only the text and its source but also its metadata which generally encapsulates data over electronic delivery. We can compare how different documents might be about the same different topics but use different languages, such as Short Message Service (SMS) and slang language used on Twitter. The Summary features allow the topics which are likely to be retweeted and the probability that it will. Topic 4 with works like “aveng”, “premier”, “world”, “ironman”, and “robertdowneyjr” are very likely to be retweeted with a P-value with 16 decimal places.