# Topic Model and Multinomial NLP Model Trained from Job Description Data

The Business School, Imperial College London
Chen, Zhiyu (CID: 02517659)
Liu, Qianru (CID: 02371934)
Liu, Yixin (CID: 02445245)
Wu, Zhongshi (CID: 02433017)

18-02-2024

# Contents

load libraries and functions . . . 3

Load job-description data . . . 4

Part 1 . . . 5

Train a twenty-topic model using the “narrative” text variable . . . 5

Store the topic model into RDS and visualize it. . . . 5

Part 2 . . . 6

Use findThoughts and labelTopics to learn what each topic is about . . . 6

Use labels to describe eight of the topics and show the five most distinctive words (by FREX) for each topic . . . 9

Part 3 . . . 10

A word cloud of the words in topic 14 . . . 10

The two documents that are estimated to have the highest proportion of the topic . . . 11

Part 4 . . . 12

Topic correlation plot . . . 12

Part 5 . . . 14

LASSO classifier model based on topic proportions feature . . . 14

Part 6 . . . 15

Fit the topic model to the test set and evaluate accuracy . . . 15

Comparison between topic model and n-gram model . . . 15

Compare the accuracy of this model to two benchmarks - word count and sentiment . . . 16

Part 7 . . . 18

Create a multinomial classifier . . . 18

The accuracy of the multinomial classifier . . . 19

The confusion matrix of the multinomial classifier . . . 20

# List of Figures

# List of Tables

## load libraries and functions

```
library(quanteda)
```

```
## Package version: 3.3.1
## Unicode version: 13.0
## ICU version: 69.1

## Parallel computing: 20 of 20 threads used.

## See https://quanteda.io for tutorials and examples.
```

```
library(ggrepel)
```

```
## Loading required package: ggplot2
```

```
library(textclean)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(sentimentr) # new one.. for sentiment
library(stm) # new one... for topic models
```

```
## stm v1.3.7 successfully loaded. See ?stm for help.
##  Papers, resources, and other materials at structuraltopicmodel.com
```

```r
library(wordcloud) # For word cloud
```

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

```r
library(igraph) # For topic correlation plot
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
##
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```r
source("TMEF_dfm.R")
source("kendall_acc.R")
```

## Load job-description data

```r
# Read the data set
jobdesc<-readRDS("cfpb_small.RDS")

# Explore main meta-data and remove every "XX", "XXX" and "XXX" in narrative column
jobdesc <- jobdesc %>%
  mutate(narrative = str_replace_all(narrative, "X+", "")) %>%
  filter(!is.na(narrative))
```

## Part 1

### Train a twenty-topic model using the "narrative" text variable

Topic modeling is a method used in machine learning and natural language processing to discover abstract topics within text. The most common algorithm for topic modeling is Latent Dirichlet Allocation (LDA).

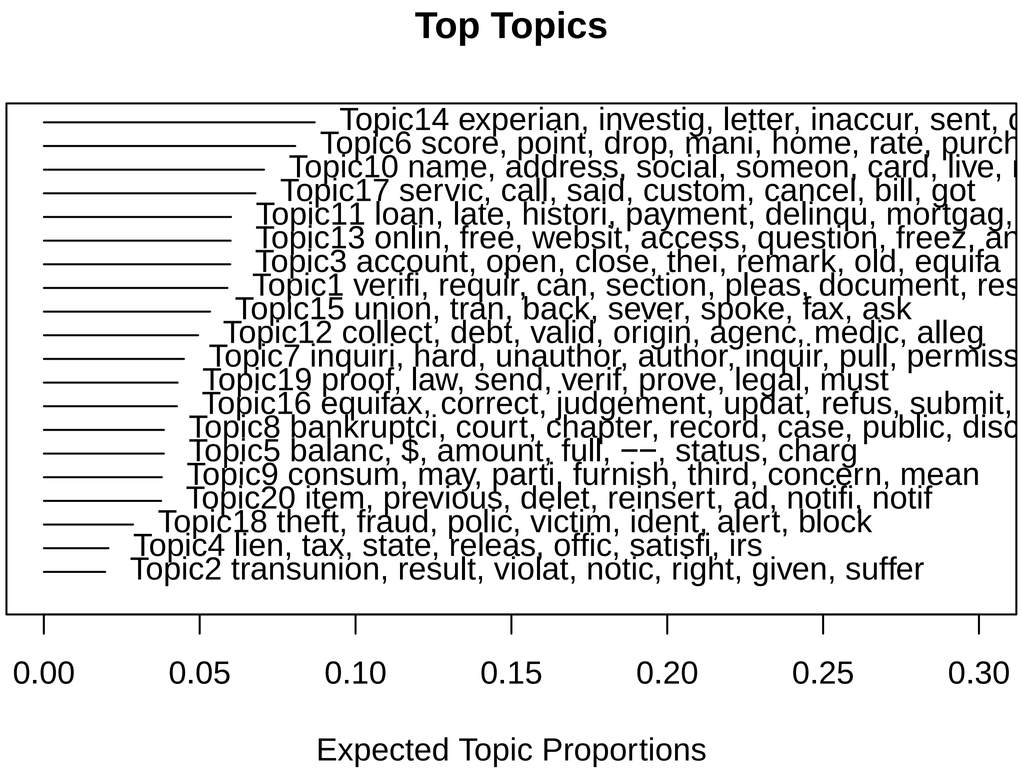
A "twenty-topic model" is a type of topic model that has been trained to identify twenty distinct topics within a collection of documents or text data.

```
# shrink the focus on the "Credit Reporting" product for topic modeling
jd_small <- jobdesc %>%
  filter(Product == "Credit reporting") %>%
  mutate(desc_wdct = str_count(narrative, "[[:alpha:]]+")) %>%
  filter(!is.na(narrative)) %>%
  mutate(sentiment = narrative %>%
           sentiment_by() %>%
           pull(ave_sentiment))

# set seed - making re-producing the same result possible
set.seed(2024)

# Training data - 0-12000 rows
# Testing data - 12000-15000 rows
train_split=sample(1:nrow(jd_small),12000)
jd_small_train<-jd_small[train_split,]
jd_small_test<-jd_small[-train_split,]

# First we need a dfm object (ngram matrix in a quanteda file format)
# Topic models are usually estimated with only unigrams, and without stopwords
jd_small_dfm_train<-TMEF_dfm(jd_small_train$narrative,ngrams=1)

jd_small_dfm_test<-TMEF_dfm(jd_small_test$narrative,ngrams=1) %>%
  dfm_match(colnames(jd_small_dfm_train))

# Train a 20-topic model
jd_topicMod20<-stm(jd_small_dfm_train,K=20)
```

### Store the topic model into RDS and visualize it.

```
# Save topic models as a RDS file
saveRDS(jd_topicMod20, file="jd_topicMod20.RDS")

# Read this RDS file to get the topic model into R environment
jd_topicMod20 <- readRDS("jd_topicMod20.RDS")

# Extract the number of topics (K) from the model's settings
topicNum = jd_topicMod20$settings$dim$K

# Since LDA doesn't assign human-readable names to topics,
# create a vector of topic names by concatenating "Topic" with the topic index.
topicNames <- paste0("Topic", 1:topicNum)
```

```
# Generate a plot summarizing the topic model.
# The plot will display the  most common words from each topic,
# with words scaled by their frequency-weighted exclusivity (frex).
plot(jd_topicMod20, type="summary", n = 7, xlim=c(0, .3), labeltype = "frex",
     topic.names = topicNames)
```



## Part 2

### Use findThoughts and labelTopics to learn what each topic is about

```
# findThoughts - Identify the most representative documents for each topic within a topic model
print("The most representative documents for topic 1:")
```

```
## [1] "The most representative documents for topic 1:"
```

```
findThoughts(model=jd_topicMod20,
             texts=jd_small_train$narrative,
             topics=1,n=1)
```

```
##
##  Topic 1:
##       I am filing this complaint because  has ignored my request to provide me with the documents tha
```

```r
cat("\n")
```

```r
cat("\n")
```

```r
# labelTopics - grab more words per topic
print("More words for the 20 topics")
```

```
## [1] "More words for the 20 topics"
```

```r
labelTopics(jd_topicMod20)
```

```
## Topic 1 Top Words:
##       Highest Prob: verifi, file, can, disput, account, document, pleas
##       FREX: verifi, requir, can, section, pleas, document, resolv
##       Lift: manner, prompt, gone, soon, ignor, mark, possibl
##       Score: manner, verifi, section, prompt, account, ignor, fcra
## Topic 2 Top Words:
##       Highest Prob: transunion, violat, result, notic, right, request, system
##       FREX: transunion, result, violat, notic, right, given, suffer
##       Lift: suffer, transunion, result, violat, code, notic, given
##       Score: suffer, transunion, violat, result, notic, right, system
## Topic 3 Top Words:
##       Highest Prob: account, report, credit, open, close, remov, show
##       FREX: account, open, close, thei, remark, old, equifa
##       Lift: thei, remark, open, account, close, duplic, asap
##       Score: thei, account, open, close, report, remov, credit
## Topic 4 Top Words:
##       Highest Prob: state, lien, tax, remov, report, credit, releas
##       FREX: lien, tax, state, releas, offic, satisfi, irs
##       Lift: lien, tax, irs, releas, satisfi, counti, offic
##       Score: lien, tax, releas, irs, state, satisfi, counti
## Topic 5 Top Words:
##       Highest Prob: $, paid, balanc, amount, full, charg, date
##       FREX: balanc, $, amount, full, --, status, charg
##       Lift: --, settlement, balanc, settl, amount, $, limit
##       Score: --, $, balanc, paid, amount, charg, full
## Topic 6 Top Words:
##       Highest Prob: credit, score, year, report, get, point, becaus
##       FREX: score, point, drop, mani, home, rate, purchas
##       Lift: fico, low, drop, lower, point, rate, score
##       Score: fico, score, credit, drop, point, rate, lower
## Topic 7 Top Words:
##       Highest Prob: credit, inquiri, report, author, remov, hard, compani
##       FREX: inquiri, hard, unauthor, author, inquir, pull, permiss
##       Lift: unauthor, inquiri, permiss, inquir, hard, employ, pull
##       Score: unauthor, inquiri, author, hard, inquir, pull, permiss
## Topic 8 Top Words:
##       Highest Prob: report, file, credit, bankruptci, record, court, case
##       FREX: bankruptci, court, chapter, record, case, public, discharg
##       Lift: chapter, discharg, bankruptci, dismiss, court, public, case
##       Score: chapter, bankruptci, court, public, discharg, dismiss, record
## Topic 9 Top Words:
```

```
##       Highest Prob: inform, consum, ani, credit, agenc, provid, state
##       FREX: consum, may, parti, furnish, third, concern, mean
##       Lift: challeng, informat, regul, third, u.s.c, procedur, entiti
##       Score: challeng, consum, furnish, inform, parti, third, may
## Topic 10 Top Words:
##       Highest Prob: name, credit, n't, address, card, number, s
##       FREX: name, address, social, someon, card, live, n't
##       Lift: ss, birth, live, social, name, els, someon
##       Score: ss, name, card, address, social, n't, secur
## Topic 11 Top Words:
##       Highest Prob: payment, report, loan, late, credit, histori, mortgag
##       FREX: loan, late, histori, payment, delinqu, mortgag, student
##       Lift: student, late, loan, histori, delinqu, miss, vehicl
##       Score: student, payment, late, loan, mortgag, delinqu, histori
## Topic 12 Top Words:
##       Highest Prob: debt, collect, valid, agenc, report, compani, credit
##       FREX: collect, debt, valid, origin, agenc, medic, alleg
##       Lift: collector, collect, medic, debt, fdcpa, alleg, valid
##       Score: collector, debt, collect, valid, agenc, owe, alleg
## Topic 13 Top Words:
##       Highest Prob: report, credit, onlin, experian, request, free, mail
##       FREX: onlin, free, websit, access, question, freez, annual
##       Lift: annual, free, onlin, websit, autom, messag, site
##       Score: annual, free, onlin, freez, websit, access, report
## Topic 14 Top Words:
##       Highest Prob: disput, experian, letter, inform, investig, sent, report
##       FREX: experian, investig, letter, inaccur, sent, certifi, disput
##       Lift: incomplet, investig, certifi, inaccuraci, letter, method, conduct
##       Score: incomplet, experian, investig, letter, disput, inaccur, sent
## Topic 15 Top Words:
##       Highest Prob: time, remov, back, ask, sever, told, contact
##       FREX: union, tran, back, sever, spoke, fax, ask
##       Lift: tran, union, handl, trade, spoken, fax, came
##       Score: tran, union, back, remov, told, ask, time
## Topic 16 Top Words:
##       Highest Prob: equifax, disput, inform, correct, report, refus, updat
##       FREX: equifax, correct, judgement, updat, refus, submit, cfpb
##       Lift: judgement, equifax, cfpb, updat, correct, judgment, refus
##       Score: judgement, equifax, correct, disput, updat, inform, refus
## Topic 17 Top Words:
##       Highest Prob: call, servic, told, said, month, bill, charg
##       FREX: servic, call, said, custom, cancel, bill, got
##       Lift: refund, cancel, rent, custom, servic, agent, apart
##       Score: refund, call, servic, bill, cancel, charg, told
## Topic 18 Top Words:
##       Highest Prob: fraud, ident, theft, report, fraudul, alert, polic
##       FREX: theft, fraud, polic, victim, ident, alert, block
##       Lift: identiti, theft, victim, polic, block, affidavit, fraud
##       Score: identiti, theft, ident, polic, fraud, victim, fraudul
## Topic 19 Top Words:
##       Highest Prob: proof, law, document, send, ask, verif, copi
##       FREX: proof, law, send, verif, prove, legal, must
##       Lift: eperian, physic, sue, prove, proof, law, demand
##       Score: eperian, proof, law, document, verif, send, signatur
## Topic 20 Top Words:
##       Highest Prob: report, item, delet, credit, remov, previous, day
##       FREX: item, previous, delet, reinsert, ad, notifi, notif
##       Lift: reinsert, notif, ad, previous, notifi, item, delet
##       Score: reinsert, delet, item, report, previous, remov, ad
```

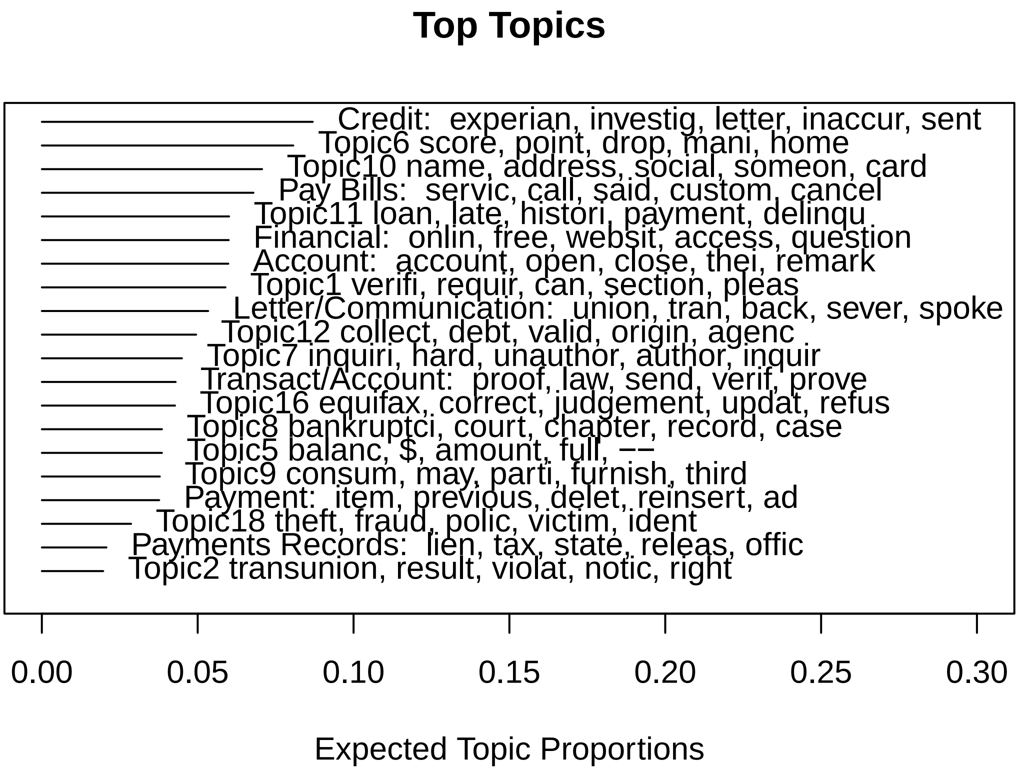
[Output Out-of-boundary Replenish]

“The most representative documents for topic 1:”

Topic 1: I am filing this complaint because has ignored my request to provide me with the documents that their company has on file that was used to verify the accounts I disputed. Being that they have gone past the 30 day mark and can not verify these accounts, under Section 611 ( 5 ) ( A ) of the FCRA - they are required to “ ” promptly delete all information which can not be verified ’’ that I have disputed. Please resolve this manner as soon as possible. Thank you.

**Use labels to describe eight of the topics and show the five most distinctive words (by FREX) for each topic**

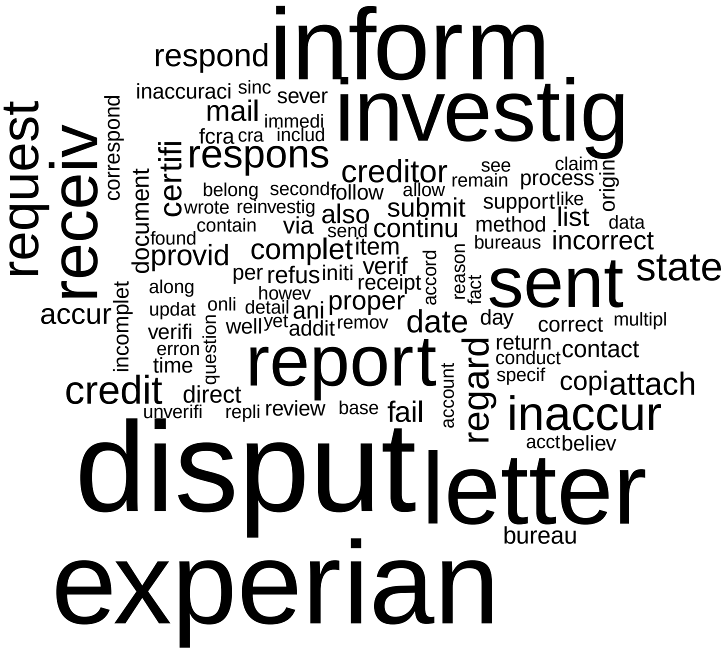
```r
topicNames[3] = "Account: "
topicNames[4] = "Payments Records: "
topicNames[15] = "Letter/Communication: "
topicNames[13] = "Financial: "
topicNames[14] = "Credit: "
topicNames[17] = "Pay Bills: "
topicNames[19] = "Transact/Account: "
topicNames[20] = "Payment: "

# Put those labels with names above on a labelTopics plot, which shows the five most distinctive words
plot(jd_topicMod20,type="summary",n = 5,xlim=c(0,.3),labeltype = "frex", topic.names = topicNames)
```



### Part 3

#### A word cloud of the words in topic 14

```r
# Word Clouds for Topic 14
cloud(jd_topicMod20, 14)
```



**The two documents that are estimated to have the highest proportion of the topic**

```
# Two documents that are estimated to have the highest proportion in topic 14
cat("Two documents that are estimated to have the highest proportion in")
```

```
## Two documents that are estimated to have the highest proportion in
```

```
findThoughts(model=jd_topicMod20,
             texts=jd_small_train$narrative,
             topics=14,n=2)
```

```
##
##  Topic 14:
##       I have I submitted  letters to the credit reporting agency asking for verification of account a
## To date I have not received any reply to by letters sent certified on   2015 and   2015 certified. Th
##      I have I submitted  letters to the credit reporting agency asking for verification of account an
## To date I have not received any reply to by letters sent certified on   2015 and   2015 certified. Th
```

[Output Out-of-boundary Replenish]

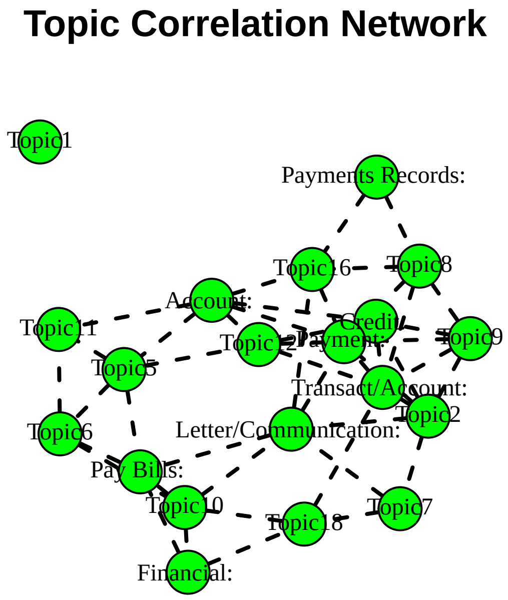
Two documents that are estimated to have the highest proportion in Topic 14: I have I submitted letters to the credit reporting agency asking for verification of account and how the verification as obtained. All letters were sent certified. I am questioning FCRA 611 nd FCRA 609 process. To date I have not received any reply to by letters sent certified on 2015 and 2015 certified. The Credit reporting agency has refused to reply and provided proper documentation for the records I listed in the correspondence and remove the items since not reply was provided. All copies of the letters and certified mail receipts are attached. I have I submitted letters to the credit reporting agency asking for verification of account and how the verification as obtained. All letters were sent certified. I am questioning FCRA 611 nd FCRA 609 process. To date I have not received any reply to by letters sent certified on 2015 and 2015 certified. The Credit reporting agency has refused to reply and provided proper documentation for the records I listed in the correspondence and remove the items since not reply was provided. All copies of the letters and certified mail receipts are attached.

## Part 4

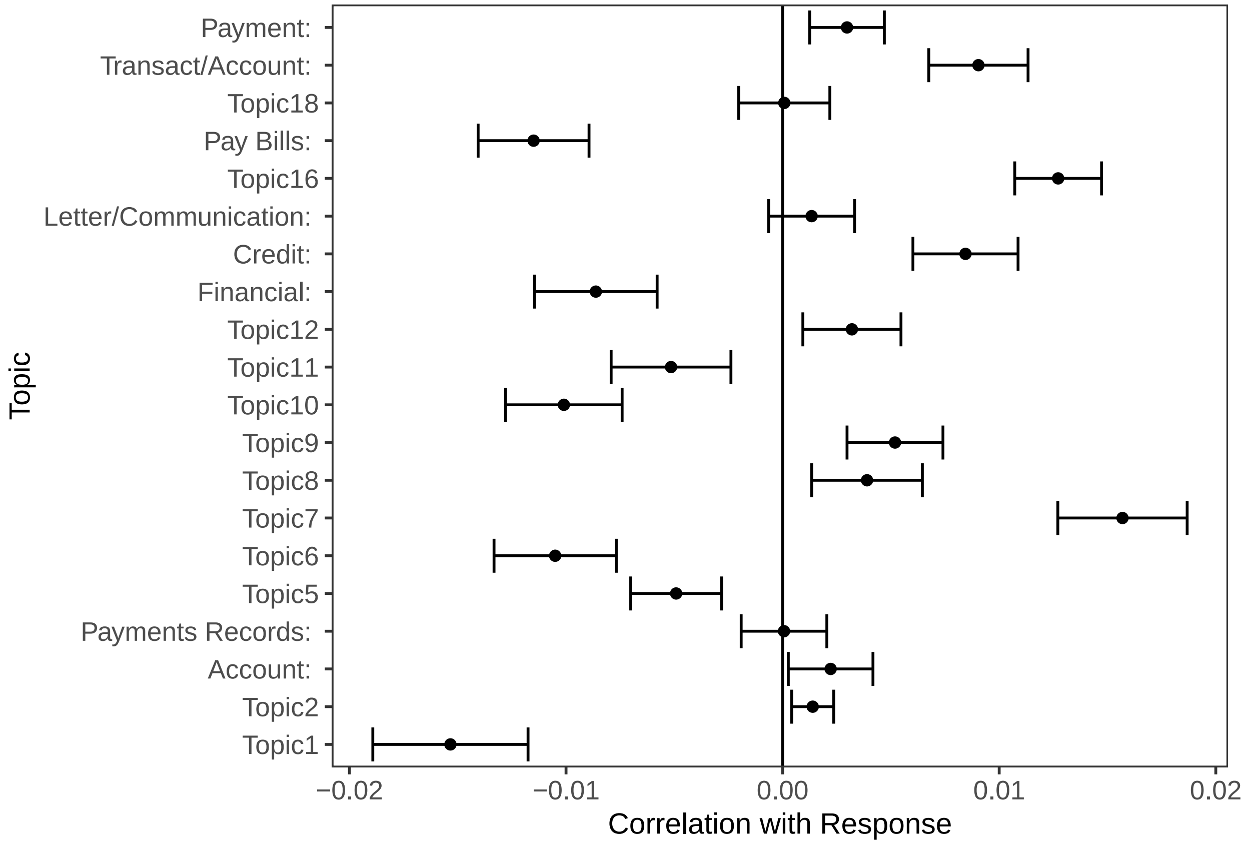
### Topic correlation plot

To find which topics correlate with each other, we would typically look for topics with similar correlation values or overlapping confidence intervals on the x-axis.

```r
# Topic correlation plot version 1
plot(topicCorr(jd_topicMod20),
     vlabels=topicNames,
     vertex.size=20,
     edge.width=2,
     edge.arrow.size=0.5,
     edge.color="black",
     main="Topic Correlation Network")
```



```r
stmEffects<-estimateEffect(1:topicNum~disputed,
                           jd_topicMod20,
                           meta= jd_small_train %>%
                             select(disputed))


# Topic correlation plot version 2
bind_rows(lapply(summary(stmEffects)$tables,function(x) x[2,1:2])) %>%
  mutate(topic=factor(topicNames,ordered=T,
                      levels=topicNames),
         se_u=Estimate+`Std. Error`,
         se_l=Estimate-`Std. Error`) %>%
  ggplot(aes(x=topic,y=Estimate,ymin=se_l,ymax=se_u)) +
  geom_point() +
  geom_errorbar() +
  coord_flip() +
  geom_hline(yintercept = 0)+
  theme_bw() +
  labs(y="Correlation with Response",x="Topic") +
  theme(panel.grid=element_blank(),
        axis.text=element_text(size=10))
```



‘Credit’ and ‘Financial’ are the two labeled topics that seem like they correlate with each other.

## Part 5

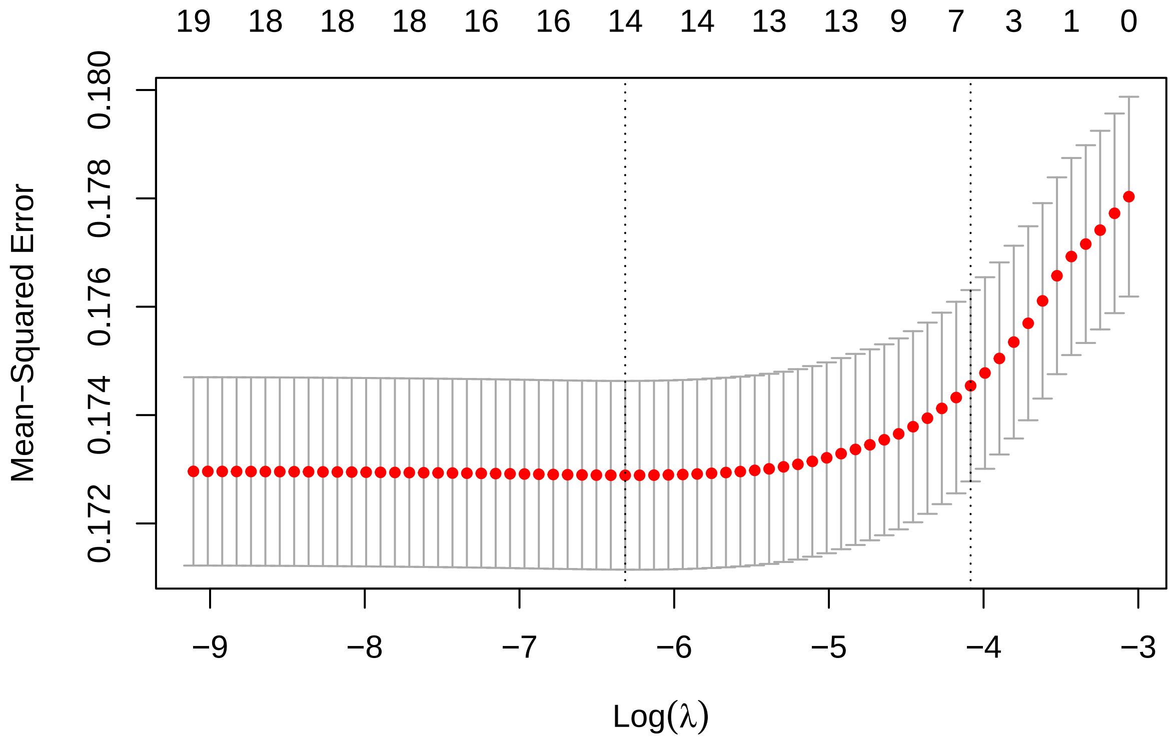
### LASSO classifier model based on topic proportions feature

Use the estimated topic proportions for each document as a feature set to train a LASSO classifier model to predict whether a company's response is disputed

```
# This contains the topic proportions for each document
topic_prop_train<-jd_topicMod20$theta

# Get the dimensions of the topic proportions
dim(topic_prop_train)
```

```
## [1] 12000    20
```

```
# Set the column names of the topic proportions to the real names of the topics for better readability.
colnames(topic_prop_train)<-topicNames

# Use these topic proportions just like any other feature
jd_model_stm<-glmnet::cv.glmnet(x=topic_prop_train,
                                y=jd_small_train$disputed)

# Note that we didn't give enough features so there is no U shape
plot(jd_model_stm)
```



```
cat("Based on the LASSO diagram, topic 15-18 are the best predictors of the outcome. Topic 10-14 & 19-20
```

```
## Based on the LASSO diagram, topic 15-18 are the best predictors of the outcome. Topic 10-14 & 19-20 a
```

## Part 6

### Fit the topic model to the test set and evaluate accuracy

```
# Fit the topic model to the test data set
topic_prop_test<-fitNewDocuments(jd_topicMod20,
                                 jd_small_dfm_test %>%
                                   convert(to="stm") %>%
                                   `$`(documents))
```

```
## ...................................................................................................
```

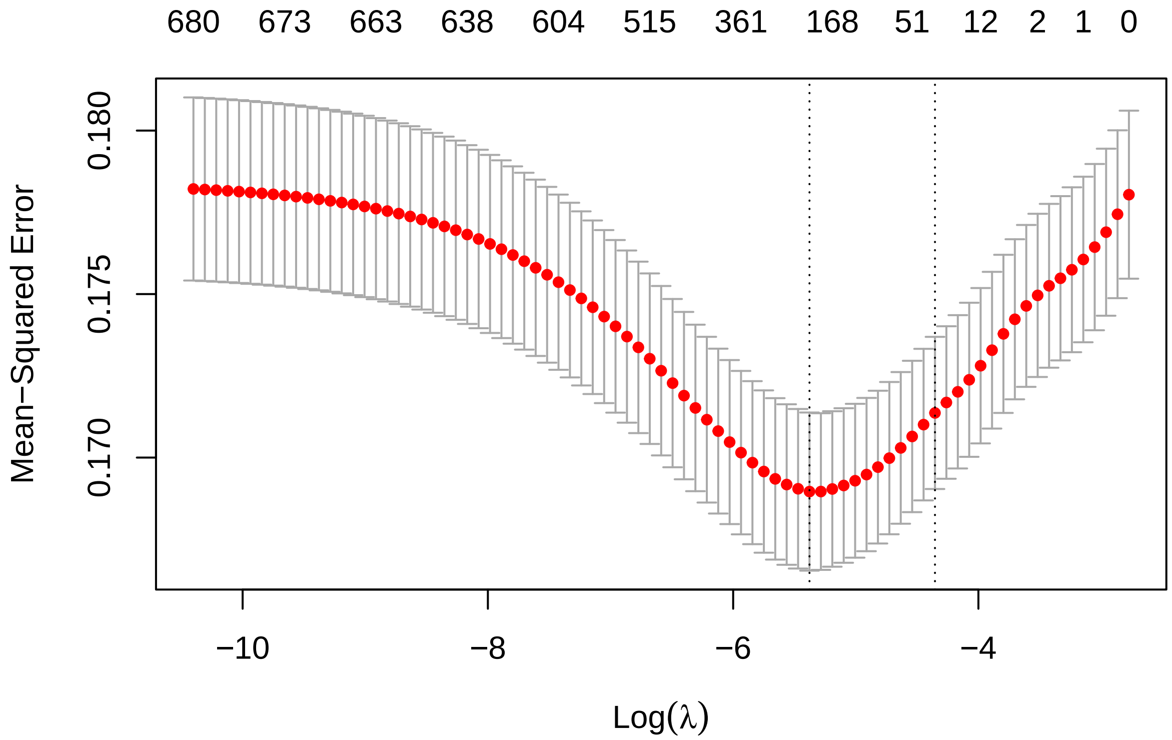
```
# Get predictions on the test data
test_stm_predict<-predict(jd_model_stm,
                          newx = topic_prop_test$theta)[,1]

# Get test accuracy
acc_stm<-kendall_acc(jd_small_test$disputed,test_stm_predict)
acc_stm
```

```
##     acc lower upper
## 1 52.56 50.78 54.35
```

The accuracy of the structural topic model is 52.56%

### Comparison between topic model and n-gram model

```
jd_model_dfm<-glmnet::cv.glmnet(x=jd_small_dfm_train,
                                y=jd_small_train$disputed)

plot(jd_model_dfm)
```



```
test_dfm_predict<-predict(jd_model_dfm,
                          newx = jd_small_dfm_test)[,1]

acc_dfm<-kendall_acc(jd_small_test$disputed,test_dfm_predict)

acc_dfm
```

```
##     acc lower upper
## 1 59.16  57.4 60.91
```

The accuracy of the document-feature matrix(n-gram model) is 59.29% Note: There is drop in performance of the topic model compared to the ngrams

**Compare the accuracy of this model to two benchmarks - word count and sentiment**

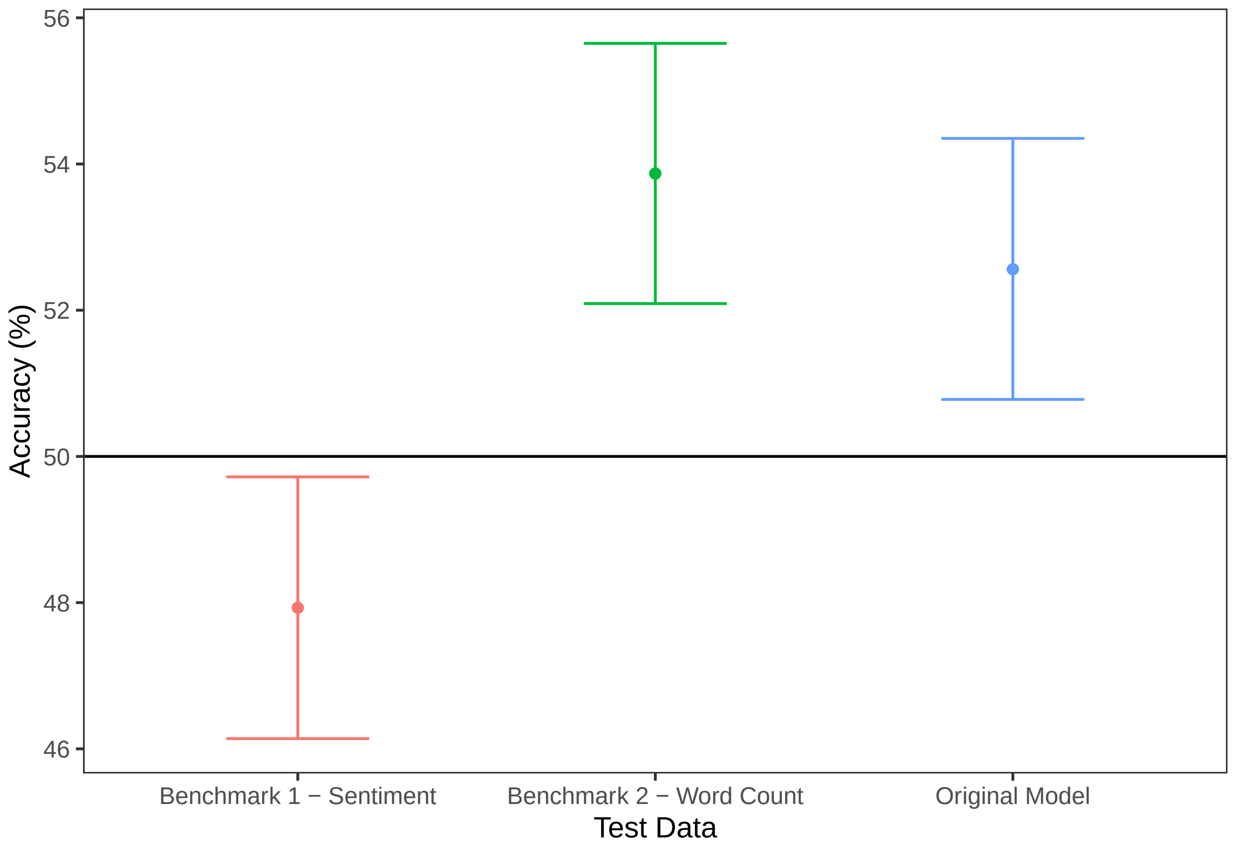
```
# Sentiment Benchmark
acc_sentiment<-kendall_acc(jd_small_test$disputed,jd_small_test$sentiment)

acc_sentiment
```

```
##     acc lower upper
## 1 47.93 46.14 49.72
```

```r
# Word-count Benchmark
acc_wdct<-kendall_acc(jd_small_test$disputed,jd_small_test$desc_wdct)

acc_wdct
```

```
##     acc lower upper
## 1 53.87 52.09 55.65
```

```r
## Combine the model accuracy and the two benchmarks' accuracy
acc_report <- bind_rows(acc_stm %>%
                          mutate(field="Original Model"),
                        acc_sentiment %>%
                          mutate(field="Benchmark 1 - Sentiment"),
                        acc_wdct %>%
                          mutate(field="Benchmark 2 - Word Count"))

acc_report %>%
  ggplot(aes(x=field,color=field,
             y=acc,ymin=lower,ymax=upper)) +
  geom_point() +
  geom_errorbar(width=.4) +
  theme_bw() +
  labs(x="Test Data",y="Accuracy (%)") +
  geom_hline(yintercept = 50) +
  theme(panel.grid=element_blank(),
        legend.position="none")
```



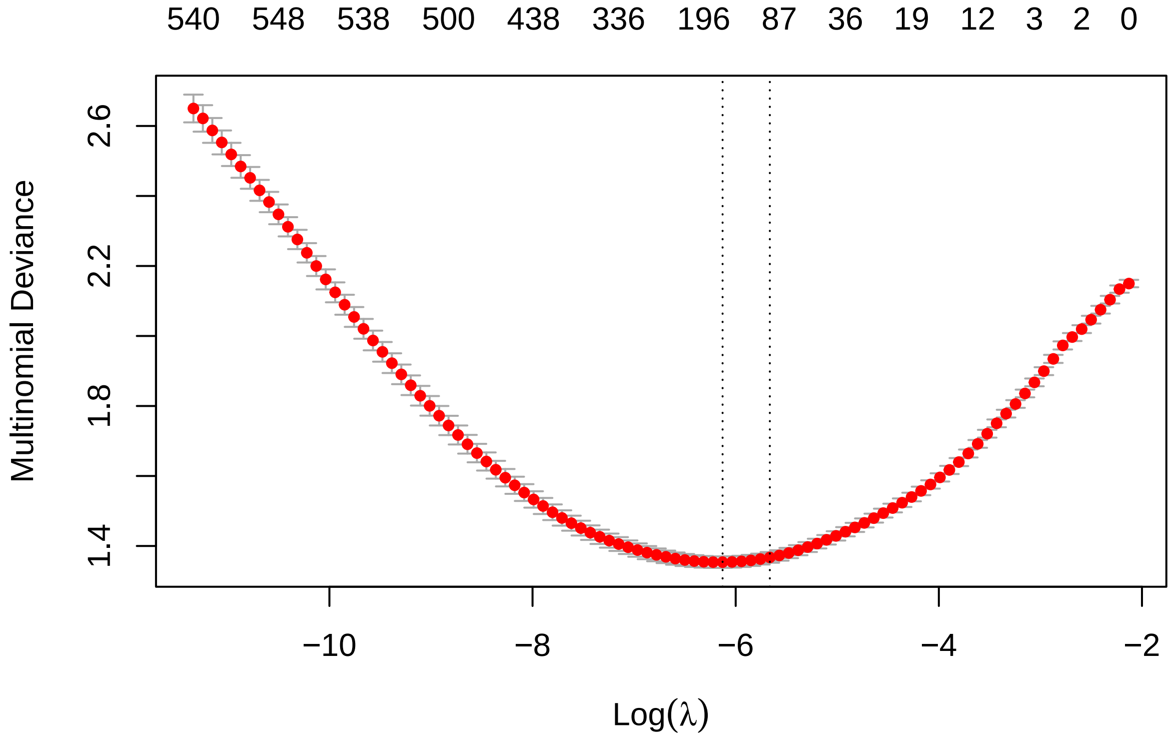
## Part 7

### Create a multinomial classifier

A multinomial classifier is a type of model used in machine learning for classification tasks that predicts the probability of each category based on a multinomial probability distribution. This kind of classifier is particularly suited for features that can occur multiple times, such as words in text data. Each document is represented as a feature vector, where features correspond to words in the vocabulary, and the values indicate the frequency of that word in the document.

Each product category has several different "Issues" in the dataset . In the training data, create a multinomial classifier to predict the five different issues from the narrative text.

```r
# Get five common categories (ranked 1-5)
topissues<-names(rev(sort(table(jobdesc$Issue))))[1:5]

# Get some descriptions from different categories
jd_issues<- jobdesc %>%
  filter(Issue%in%topissues  & !is.na(Issue))

# Set seed to make the result repeatable
set.seed(2024)

# Split the dataset
# Training data - 1-12000 rows
# Testing data - 12000-15000 rows
train_split=sample(1:nrow(jd_issues), 12000)
jd_issues_train<-jd_issues[train_split,]
jd_issues_test<-jd_issues[-train_split,]

# Feature extraction (same as n-grams)
jd_issues_dfm_train<-TMEF_dfm(jd_issues_train$narrative,ngrams=1)

jd_issues_dfm_test<-TMEF_dfm(jd_issues_test$narrative,
                             ngrams=1,min.prop=0) %>%
  dfm_match(colnames(jd_issues_dfm_train))

# Plot multinational classifier's LASSO
jd_model_issues<-glmnet::cv.glmnet(x=jd_issues_dfm_train,
                                   y=jd_issues_train$Issue,
                                   family="multinomial")

plot(jd_model_issues)
```



### The accuracy of the multinomial classifier

```r
# Type="class" - get a single predicted label for each document
# Type="response" - get a probability that each document is in each class
# Too much output for response so won't show here
issues_predict_label<-predict(jd_model_issues,
                              newx = jd_issues_dfm_test,
                              type="class")[,1]

# raw accuracy
mean(issues_predict_label==jd_issues_test$Issue)
```

```
## [1] 0.7466667
```

### The confusion matrix of the multinomial classifier

```
# The confusion matrix
table(issues_predict_label,substr(jd_issues_test$Issue,0,10))
```

```
##
## issues_predict_label                       Credit mon Credit rep Improper u
##   Credit monitoring or identity protection         44          1          1
##   Credit reporting company’s investigation          2        275          3
##   Improper use of my credit report                  2          3         45
##   Incorrect information on credit report           37        407         83
##   Unable to get credit report/credit score         17         10          3
##
## issues_predict_label                       Incorrect  Unable to
##   Credit monitoring or identity protection          3          3
##   Credit reporting company’s investigation         88          7
##   Improper use of my credit report                  5          0
##   Incorrect information on credit report         1722         70
##   Unable to get credit report/credit score         15        154
```

```
# Output the confusion matrix csv
table(issues_predict_label,jd_issues_test$Issue) %>%
  write.csv("issues_table.csv")
```

“Credit monitoring or identity protection”: 44 instances were correctly predicted as “Credit monitoring or identity protection”. However, 1 was incorrectly predicted as “Credit reporting company’s investigation”, 1 as “Improper use of my credit report”, 3 as “Incorrect information on credit report”, and 3 as “Unable to get credit report/credit score”. Overall 44/52 are correct (84.6%).

“Credit reporting company’s investigation”: 275 instances were correctly predicted as “Credit reporting company’s investigation”. However, 2 were incorrectly predicted as “Credit monitoring or identity protection”, 45 as “Improper use of my credit report”, 8 as “Incorrect information on credit report”, and 7 as “Unable to get credit report/credit score”. Overall 275/375 are correct (73.3%)

“Improper use of my credit report”: 45 instances were correctly predicted as “Improper use of my credit report”. However, 2 were incorrectly predicted as “Credit monitoring or identity protection”, 3 as “Credit reporting company’s investigation”, 5 as “Incorrect information on credit report”. Overall 45/55 are correct (81.8%).

“Incorrect information on credit report”: 1722 instances were correctly predicted as “Incorrect information on credit report”. However, 37 were incorrectly predicted as “Credit monitoring or identity protection”, 407 as “Credit reporting company’s investigation”, 83 as “Improper use of my credit report”, 70 as “Unable to get credit report/credit score”. Overall 1722/2319 are correct (74.3%).

“Unable to get credit report/credit score”: 154 instances were correctly predicted as “Unable to get credit report/credit score”. However, 17 were incorrectly predicted as “Credit monitoring or identity protection”, 10 as “Credit reporting company’s investigation”, 3 as “Improper use of my credit report”, 15 as “Incorrect information on credit report”. Overall 154/199 are correct (77.4%)

Thus, the model was most likely to make mistake on “Credit reporting company’s investigation”.