WARWICK BUSINESS SCHOOL

Masters Programmes

Assignment Cover Sheet

Submitted by:	***1992799***
Date Sent:	**7th May 2020**
Module Title:	**Text Analytics**
Module Code:	**IB9CW0**
Date/Year of Module:	**2019/2020**
Submission Deadline:	**30th April 2020 (extension to 7th May 2020)**
Word Count:	**9160**
Number of Pages:	**70**
Question:	

The goal of this assignment is to familiarize you with the complete process of *extracting*, *refining* and *delivering* insights of particular financial value that are extracted from unstructured data of non-conventional size from company reports.

"I declare that this work is entirely my own in accordance with the University's Regulation 11 and the WBS guidelines on plagiarism and collusion. All external references and sources are clearly acknowledged and identified within the contents.

No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has been done it may result in me being reported for self-plagiarism and an appropriate reduction in marks may be made when marking this piece of work."

Preface

This study is conducted to fulfill the individual assignment requirement for Text Analytics module. Subject of the study is Security and Exchange Commision (SEC) annual (10-K) and quarterly (10-Q) reports of S&P 500 companies between 2009 and 2019. This document is broken down into 3 major parts; (A) the construction of a corpus, (B) sentiment analysis, and (C) topic modelling.

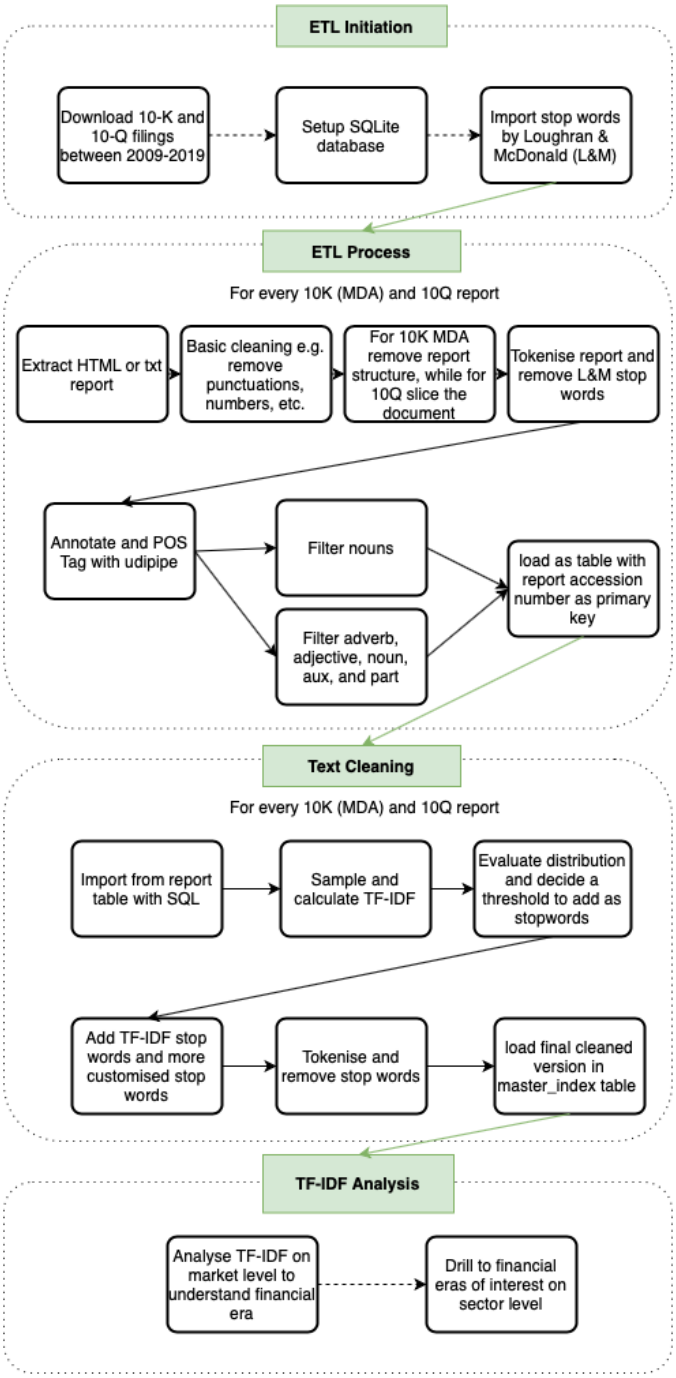
Each part, will start with a visual representation of the process workflow, then followed by the main content that includes chunks of R codes and graphs, where eventually it is closed with a conclusion section.

Please note that author believes that the highlights of the results are mainly encaptured in graphs with only brief description where necessary. Description of graphs may include justification of approach or main findings. Therefore, the conclusion section will only be used to summarise the findings where the length is kept to the minimum.

Notes to Papa Nikos — I really had fun doing the assignment, and hope you will be proud of our results. Thank you!

Part A — Construction of Corpus

Process Workflow



ETL Initiation

This section aims to set up the environment to conduct the analysis.

R Packages

```
knitr::opts_knit$set(root.dir = '/Volumes/Buku Gibran/edgar')
knitr::opts_chunk$set(eval = FALSE)
library(tidyverse)
library(edgar)
library(XML)
library(lubridate)
library(tm)
library(RSQLite)
library(tidytext)
library(udpipe)
library(rvest)

library(readxl)
library(qdap)
library(sentimentr)
library(textfeatures)
library(BatchGetSymbols)

library(lubridate)
library(DataExplorer)
library(gridExtra)
library(stm)
```

Download Financial Reports

```
# ----- The S&P 500
sp500 <- read_csv('sp500.csv')

# ----- Download MDA chapter 10-K and HTML of 10-Q
edgar::getMgmtDisc(cik.no = sp500$CIK, filing.year = c(2009:2019))
edgar::getFilingsHTML(cik.no = sp500$CIK, form.type = '10-Q', filing.year = c
(2009:2019))
```

Due to the lack computing resources, and the short timeframe, this study only considers the management and discussion (MD&A) section for 10-K and the whole report for 10-Q. However, Brown and Tucker (2010) reported that MD&A section drives reaction to 10-K filing, thus affecting the financial market reaction. Therefore, justifying that MD&A should be able to represent the content of the full 10-K report.

Setup SQLite database

```
# ----- Initiate SQLite database
conn <- dbConnect(RSQLite::SQLite(), "edgar.db")
dbWriteTable(conn,"sp500", sp500) # create sp500 table for information about
S&P500
rm(sp500)

# ----- Insert Master Indexes to db
master_index_list <-list.files('Master Indexes')
for(i in 1:length(master_index_list)) {

  load(paste0('Master Indexes/',master_index_list[i]))

  local_df <- year.master %>%
    filter(cik %in% sp500$CIK, form.type %in% c('10-Q', '10-K')) %>%
    mutate(date.filed = as.Date(date.filed)) %>%
    mutate(year_filed = year(date.filed)) %>%
    mutate(accession.number = gsub(".*/", "", edgar.link)) %>%
    mutate(accession.number = gsub('.txt','',accession.number)) %>%
    select(-edgar.link)

  colnames(local_df) <- gsub("\\.", "_", colnames(local_df)) # column names w
ith dots (.) will confuse SQL

  dbWriteTable(conn,"master_index", local_df, append = TRUE) # create master_
index Table
}

rm(local_df, year.master)

# ----- Check Records
dbGetQuery(conn, 'SELECT count(cik) from master_index') # 20678
dbGetQuery(conn, 'SELECT count(cik) from sp500') # 505

dbDisconnect(conn)
```

The final SQL schema

```
1.  BEGIN TRANSACTION;
2.  DROP TABLE IF EXISTS "sentiment";
3.  CREATE TABLE IF NOT EXISTS "sentiment" (
4.      "accession_number"  TEXT,
5.      "LM_total_words"    INTEGER,
6.      "LM_sent"   REAL,
7.      "LM_positive"   REAL,
8.      "LM_negative"   REAL,
9.      "LM_uncertainty"    REAL,
10.     "LM_litigious"  REAL,
```

```
    "LM_constraining"   REAL,
    "LM_strong_modal"   REAL,
    "LM_weak_modal" REAL,
    "complexity"    REAL,
    "sentimentr"    REAL,
    "afinn_sent"    REAL,
    "bing_total_words"  INTEGER,
    "bing_sent" REAL,
    "bing_positive" REAL,
    "bing_negative" REAL,
    "nrc_total_words"   INTEGER,
    "nrc_sent"  REAL,
    "nrc_positive"  REAL,
    "nrc_negative"  REAL,
    "nrc_anger" REAL,
    "nrc_fear"  REAL,
    "nrc_trust" REAL,
    "nrc_sadness"   REAL,
    "nrc_surprise"  REAL,
    "nrc_disgust"   REAL,
    "nrc_joy"   REAL,
    "nrc_anticipation"  REAL,
    "sent_syuzhet"  REAL,
    "sent_vader"    REAL
);
DROP TABLE IF EXISTS "sp500";
CREATE TABLE IF NOT EXISTS "sp500" (
    "symbol"    TEXT,
    "security"  TEXT,
    "gics_sector"   TEXT,
    "gics_sub_industry" TEXT,
    "cik"   TEXT
);
DROP TABLE IF EXISTS "master_index";
CREATE TABLE IF NOT EXISTS "master_index" (
    "cik"   TEXT,
    "company_name"  TEXT,
    "form_type" TEXT,
    "date_filed"    REAL,
    "quarter"   INTEGER,
    "year_filed"    REAL,
    "accession_number"  TEXT,
    "cleaned_text"  TEXT,
    "cleaned_noun"  TEXT,
    "return_adjusted_price" double,
    "price_adjusted_ratio"  double
);
COMMIT;
```

Import Stop Words from Loughran & McDonald's

Stop words are taken from Loughran & McDonald's official resource website of the University of Notre Dame. Edgar package's getSentiment function is built on top of the listed stop words. Manual import is used for more flexibility compared to using getSentiment function.

Link: https://sraf.nd.edu/textual-analysis/resources/

```
# ----- Stopwords by Loughran McDonald
stopw_loughran_mcdonald <- c()
stopw_dictionaries <-list.files('stopw_loughran_mcdonald')

for (i in 1:length(stopw_dictionaries)) {
    file_path <-paste('stopw_loughran_mcdonald', stopw_dictionaries[i],sep="/
")
    local_list <- read_lines(file_path)
    local_list <- iconv(local_list, "ASCII", "UTF-8", sub="") %>% tolower()
    stopw_loughran_mcdonald <- c(stopw_loughran_mcdonald, local_list)
}

# ----- Add customised stopwords
stopw_custom <- c('vs', 'financial', 'statement', 'exhibit','report','figure'
,'fig','tab','table', 'mda', 'company', 'footnote', 'page')


# ----- Finalising Stopwords
stopw_final <- c(stopw_loughran_mcdonald, stopw_custom)
rm(stopw_loughran_mcdonald, stopw_custom, stopw_dictionaries, file_path, loca
l_list)

# ----- Get the udpipe model
ud_model <- udpipe_download_model(language = "english", overwrite = F)
ud_model <- udpipe_load_model(ud_model$file_model)
```

ETL Process

Extracting, transforming, and loading data is major part of the study, as it is the backbone of providing insightful results. This section's goal is to extract textual information from the reports and store them in a relational schema.

Process MD&A chapter 10K reports

```
conn <- dbConnect(RSQLite::SQLite(), "edgar.db")

the_index <- dbGetQuery(conn, 'SELECT accession_number FROM master_index WHER
E form_type = "10-K" ORDER BY date_filed')

split_size <- 50
split_the_index <- split(the_index$accession_number, ceiling(seq_along(the_in
dex$accession_number)/split_size))

rm(the_index)

for (s in 1:length(split_the_index)) {
file_pattern <- paste0(split_the_index[[s]], '.txt')
```

```r
listed_files <- list.files('MD&A section text', pattern = paste0(file_pattern
, collapse = "|"))
file_path <- paste0('MD&A section text/', listed_files)

rm(file_pattern, listed_files)

for(i in 1:length(file_path)) {

  # ----- Clean Text
  text_file <- read_lines(file_path[i])
  text_transformed <- tibble(
              company_name = tolower(gsub('Company Name: ','',text_file[2]))
,
              accession_number = gsub('Accession Number: ','',text_file[5]),
              mgmtdisc = gsub(" s "," ",tolower(text_file[8]) %>%
                              removePunctuation()) %>%
                              removeNumbers() %>%
                              stripWhitespace())

  rm(text_file)

  company_name <- unlist(str_split(text_transformed$company_name, " ", n= nch
ar(text_transformed$company_name)))[1]

  sub_this <- c("item","management", "managements", "discussion and analysis"
, "financial condition", "results of operations",
                company_name)
  text_transformed$cleaned <- gsub(paste0(sub_this, collapse = '|'),"", text_
transformed$mgmtdisc)

  rm(company_name, sub_this)

  # ----- Tokenisation and Part-of-speech Tagging
  tokenised <- text_transformed %>%
    select(accession_number, cleaned) %>%
    unnest_tokens(word, cleaned) %>%
    group_by(accession_number, word) %>%
    filter(!word %in% stopw_final)

    rm(text_transformed)

      # Udpipe Annotating
      local_df <- udpipe_annotate(tokenised$word,
                        doc_id = tokenised$accession_number,
                        object = ud_model) %>% as.data.frame()
      rm(tokenised)

      # Get nouns only
      annotated_nouns <- local_df %>%
```

```
        filter(upos == "NOUN") %>%
        select(doc_id,lemma) %>%
        group_by(doc_id) %>%
        summarise(cleaned_noun = paste(lemma, collapse = " ")) %>%
        rename(accession_number = doc_id)

      # Get the most important POS
      annotated_full <- local_df %>%
        filter(upos %in% c("ADV","ADJ","NOUN", "AUX", "PART")) %>%
        select(doc_id,lemma) %>%
        group_by(doc_id) %>%
        summarise(cleaned_text = paste(lemma, collapse = " ")) %>%
        rename(accession_number = doc_id)

      # Store the data into lists we created before for loop
      local_df <- annotated_nouns %>%
      left_join(annotated_full, by= 'accession_number')

  # ----- Insertion to SQL table
  dbWriteTable(conn,"cleaned_10k_mda", local_df, append = TRUE) # create staging Table

  temp_report <- dbGetQuery(conn, paste0('SELECT cik, company_name, year_filed, form_type, cleaned_10k_mda.accession_number
                          FROM master_index LEFT JOIN cleaned_10k_mda ON cleaned_10k_mda.accession_number = master_index.accession_number
                          WHERE cleaned_10k_mda.accession_number = "',local_df$accession_number[1],'"'))

  print(paste(temp_report$form_type, temp_report$year_filed, 'report for CIK:',temp_report$cik, temp_report$company_name, 'has been processed.' ))

  rm(annotated_nouns, annotated_full, local_df, temp_report)
  }
}
rm(file_path)
```

Process MD&A chapter 10-Q reports

After reading through a portion of 10-Q reports, author identified that most quarterly reports follow the same notations indicating chapters, where the main content of the report stretches from part 1 item 3, the MD&A section, all the way to content before the exhibits. The 'notes to….' Text can be used to slice the document cleanly from the cover sheet to the content before the MD&A section as it appears in the section but not in the table of content.

```
the_index <- dbGetQuery(conn, 'SELECT cik, accession_number FROM master_index WHERE form_type = "10-Q" ')
z = 0
```

```r
split_the_index <- split(the_index, with(the_index, interaction(cik)), drop =
TRUE)
rm(the_index)


for (s in 1:length(split_the_index)) {
file_pattern <- paste0(split_the_index[[s]]$accession_number, '.html')
listed_files <- list.files(paste0('Edgar filings_HTML view/Form 10-Q/',split_
the_index[[s]]$cik[1]), pattern = paste0(file_pattern, collapse = "|"))
accession_number <- split_the_index[[s]]$accession_number
file_path <- paste0('Edgar filings_HTML view/Form 10-Q/',split_the_index[[s]]
$cik[1],'/', listed_files)

rm(file_pattern, listed_files)

  for(i in 1:length(file_path)) {
  doc <- read_html(file_path[i],  options = "HUGE") %>%
    html_text() %>%
    tolower() %>%
    removePunctuation() %>%
    removeNumbers %>%
    stripWhitespace()

  doc_begin <- regmatches(doc,gregexpr("(?<=notes to).*",doc,perl=TRUE))[[1]]
# regex to slice document from chapter notes on financial statements
  if(length(doc_begin) > 0) {doc_begin <- doc_begin} else {doc_begin <- regma
tches(doc,gregexpr("(?<=notes).*",doc,perl=TRUE))[[1]]}
  if(length(doc_begin) > 0) {doc_begin <- doc_begin} else {doc_begin <- regma
tches(doc,gregexpr("(?<=part).*",doc,perl=TRUE))[[1]]}
  doc <- regmatches(doc_begin,gregexpr(".*(?<=item exhibits)",doc,perl=TRUE))
[[1]][1] # regex to slice document until chapter "exhibits"
  rm(doc_begin)

  sub_this <- c("table of contents",
                "page",
                "item",
                "part",
                "financial information",
                "financial statements",
                "summary of significant accounting policies",
                "(continued)",
                "financial intstruments",
                "derivatives and hedging activities",
                "fair value hedges",
                "fair value measurements",
                "results of operations",
                "financial statements",
                "consolidated statements",
                "consolidated financial",
```

```
                "notes to consolidated financial statements",
                "quantitative and qualitative disclosure about market risk",
                "controls and procedures",
                "other information",
                "legal proceedings",
                "risk factors",
                "unregistered sales of equity securities and use of proceeds"
,
                "management discussion and analysis of financial condition an
d results of operations",
                "managements discussion and analysis of financial condition a
nd results of operations",
                "exhibits") #company_name
  doc <- gsub(paste0(sub_this, collapse = '|')," ", doc)

  text_transformed <- tibble(accession_number = accession_number[i], cleaned
= doc)
  rm(doc)

  # ----- Tokenisation and Part-of-speech Tagging
    tokenised <- text_transformed %>%
      select(accession_number, cleaned) %>%
      unnest_tokens(word, cleaned) %>%
      group_by(accession_number, word) %>%
      filter(!word %in% stopw_final)

      rm(text_transformed)

        # Udpipe Annotating
        local_df <- udpipe_annotate(tokenised$word,
                          doc_id = tokenised$accession_number,
                          object = ud_model) %>% as.data.frame()
        rm(tokenised)

        # Get nouns only
        annotated_nouns <- local_df %>%
          filter(upos == "NOUN") %>%
          select(doc_id,lemma) %>%
          group_by(doc_id) %>%
          summarise(cleaned_noun = paste(lemma, collapse = " ")) %>%
          rename(accession_number = doc_id)

        # Get the most important POS
        annotated_full <- local_df %>%
          filter(upos %in% c("ADV","ADJ","NOUN", "AUX", "PART")) %>%
          select(doc_id,lemma) %>%
          group_by(doc_id) %>%
          summarise(cleaned_text = paste(lemma, collapse = " ")) %>%
          rename(accession_number = doc_id)
```

```
        # Store the data into lists we created before for loop
        local_df <- annotated_nouns %>%
        left_join(annotated_full, by= 'accession_number')

   # ----- Insertion to SQL table
  dbWriteTable(conn,"cleaned_10q", local_df, append = TRUE) # create master_i
ndex Table

  temp_report <- dbGetQuery(conn, paste0('SELECT cik, quarter ,company_name,
year_filed, form_type, cleaned_10q.accession_number
                          FROM master_index LEFT JOIN cleaned_10q ON cleaned_1
0q.accession_number = master_index.accession_number
                          WHERE cleaned_10q.accession_number = "',local_df$acc
ession_number[1],'"'))

  print(paste(temp_report$form_type, temp_report$quarter, temp_report$year_fi
led, 'report for CIK:',temp_report$cik, temp_report$company_name, 'has been p
rocessed.' ))

  rm(annotated_nouns, annotated_full, local_df, temp_report)
  }
z = z + 1
print(paste(z, "out of 499"))

}
```

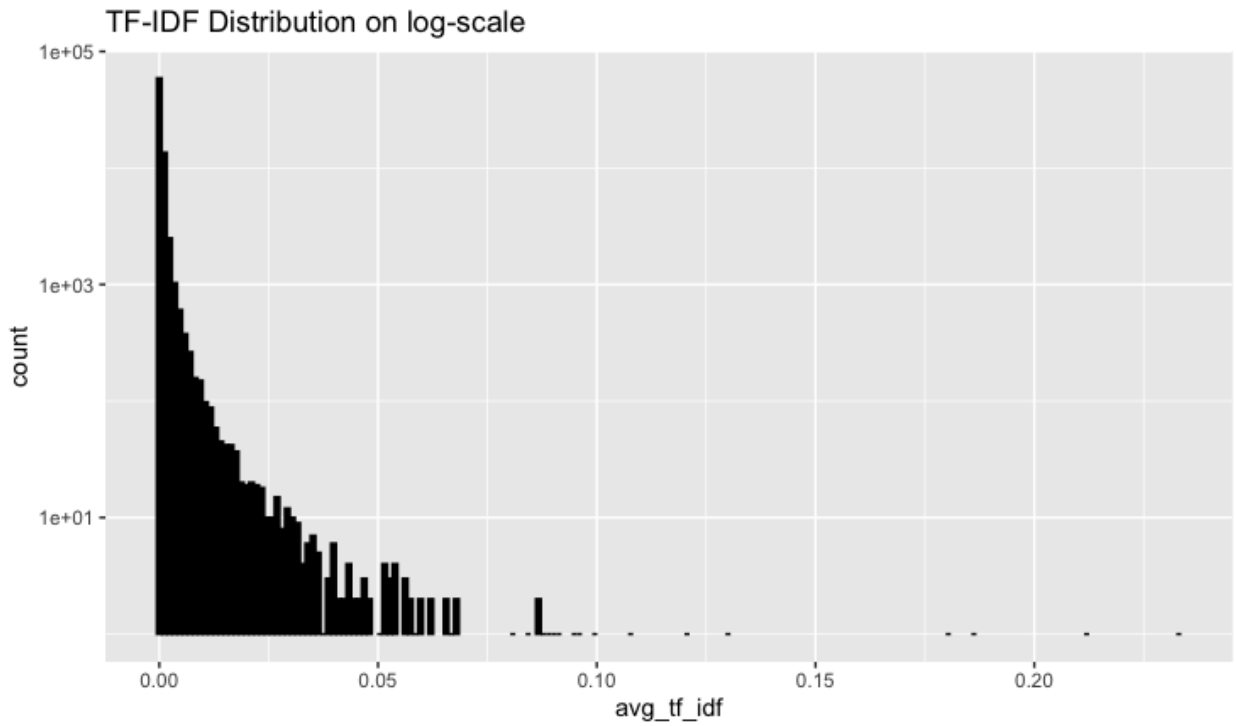
Text Cleaning

Once the reports are loaded in SQLite database where every row represents a report, this section aims to cleanse and ensure the text are ready for analysis. It can also be viewed as the 2nd iteration of transformation of the ETL process.

TF-IDF Stopwords Identification

```
# ----- Sample on S&P Market 2011 and 2017
conn <- dbConnect(RSQLite::SQLite(), "edgar.db")
sample_reports <- dbGetQuery(conn, 'SELECT cik, cleaned_noun FROM master_inde
x WHERE year_filed IN (2011, 2017)')

# ----- Calculate TF-IDF
tf_idf_samples <- sample_reports %>%
  unnest_tokens(word, cleaned_noun) %>%
  count(cik, word, sort = TRUE) %>%
  ungroup() %>%
  bind_tf_idf(word, cik, n)
```

```r
rm(sample_reports)

summarised_tf_idf <- tf_idf_samples %>%
  group_by(word) %>%
  summarise(avg_tf_idf = mean(tf_idf)) %>%
  arrange(desc(avg_tf_idf))

rm(tf_idf_samples)

# ----- Plotting TF-IDF Distribution
ggplot(summarised_tf_idf, aes(x=avg_tf_idf)) +
  geom_histogram(color="black", fill="black", bins = 200) +
  scale_y_log10() +
  labs(title = "TF-IDF Distribution on log-scale")


# ----- Adding buttom 10% words with the lowest TF-IDF as Stopwords
top_90_percent <- summarised_tf_idf %>%
  top_frac(0.90) %>%
  arrange(desc(avg_tf_idf))

buttom_10_percent <- summarised_tf_idf %>%
  anti_join(top_90_percent) %>%
  arrange(desc(avg_tf_idf))

stopw_tfidf <- buttom_10_percent$word
stopw_final <- c(stopw_final, stopw_tfidf)

rm(stopw_tfidf, summarised_tf_idf, top_90_percent, buttom_10_percent)

dbExecute(conn, 'ALTER TABLE master_index ADD COLUMN cleaned_text TEXT;') # a
dd cleaned_text to master_index
dbExecute(conn, 'ALTER TABLE master_index ADD COLUMN cleaned_noun TEXT;') # a
dd clean noun column to master_index
```



TF-IDF of the words follow a zipf's law distribution, and after some investigation of different cut-off percentage it was decided to cut-off the buttom 10% as stop-words with a total of almost 3000 words.

Tokenise and Remove Stop Words

```
# ------ 10-K MDA Text Cleaning
conn <- dbConnect(RSQLite::SQLite(), "edgar.db")


market_level <- dbGetQuery(conn, 'SELECT company_name, master_index.accession
_number, cleaned_10k_mda.cleaned_text, cleaned_10k_mda.cleaned_noun
           FROM master_index INNER JOIN cleaned_10k_mda ON master_index.acces
sion_number = cleaned_10k_mda.accession_number ')

for(i in 1:nrow(market_level)){
  tryCatch({
    x <- market_level[i,]
    name <- strsplit(tolower(x[1]), " ")[[1]] # identify name of the company

    cleaned_noun <- x %>%
      unnest_tokens(word, cleaned_noun) %>%
      filter(!word %in% name) %>% # remove company name
      filter(!word %in% stopw_final) %>% # remove stopwords
      summarise(cleaned_noun = paste(word, collapse = " ")) # combine tokens
to store
```

```
    cleaned_text <- x %>%
      unnest_tokens(word, cleaned_text) %>%
      filter(!word %in% name) %>%
      filter(!word %in% stopw_final) %>%
      summarise(cleaned_text = paste(word, collapse = " "))

    accession_number <- x$accession_number[1]


    dbExecute(conn, paste0("UPDATE master_index SET cleaned_text = '",cleaned
_text ,"', cleaned_noun = '",cleaned_noun ,"' WHERE accession_number = '",acc
ession_number ,"'"))
    rm(x, name, cleaned_noun, cleaned_text, accession_number)

    print(paste(i, "of", nrow(market_level), "10K MDA"))

  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}

# ----- 10-Q Text Cleaning
market_level <- dbGetQuery(conn, 'SELECT company_name, master_index.accession
_number, cleaned_10q.cleaned_text, cleaned_10q.cleaned_noun
           FROM master_index INNER JOIN cleaned_10q ON master_index.accession
_number = cleaned_10q.accession_number ')

for(i in 1:nrow(market_level)){
  tryCatch({
  x <- market_level[i,]
  name <- strsplit(tolower(x[1]), " ")[[1]]

  cleaned_noun <- x %>%
    unnest_tokens(word, cleaned_noun) %>%
    filter(!word %in% name) %>%
    filter(!word %in% stopw_final) %>%
    summarise(cleaned_noun = paste(word, collapse = " "))

  cleaned_text <- x %>%
    unnest_tokens(word, cleaned_text) %>%
    filter(!word %in% name) %>%
    filter(!word %in% stopw_final) %>%
    summarise(cleaned_text = paste(word, collapse = " "))

  accession_number <- x$accession_number[1]


  dbExecute(conn, paste0("UPDATE master_index SET cleaned_text = '",cleaned_t
ext ,"', cleaned_noun = '",cleaned_noun ,"' WHERE accession_number = '",acces
sion_number ,"'"))
  rm(x, name, cleaned_noun, cleaned_text, accession_number)
```

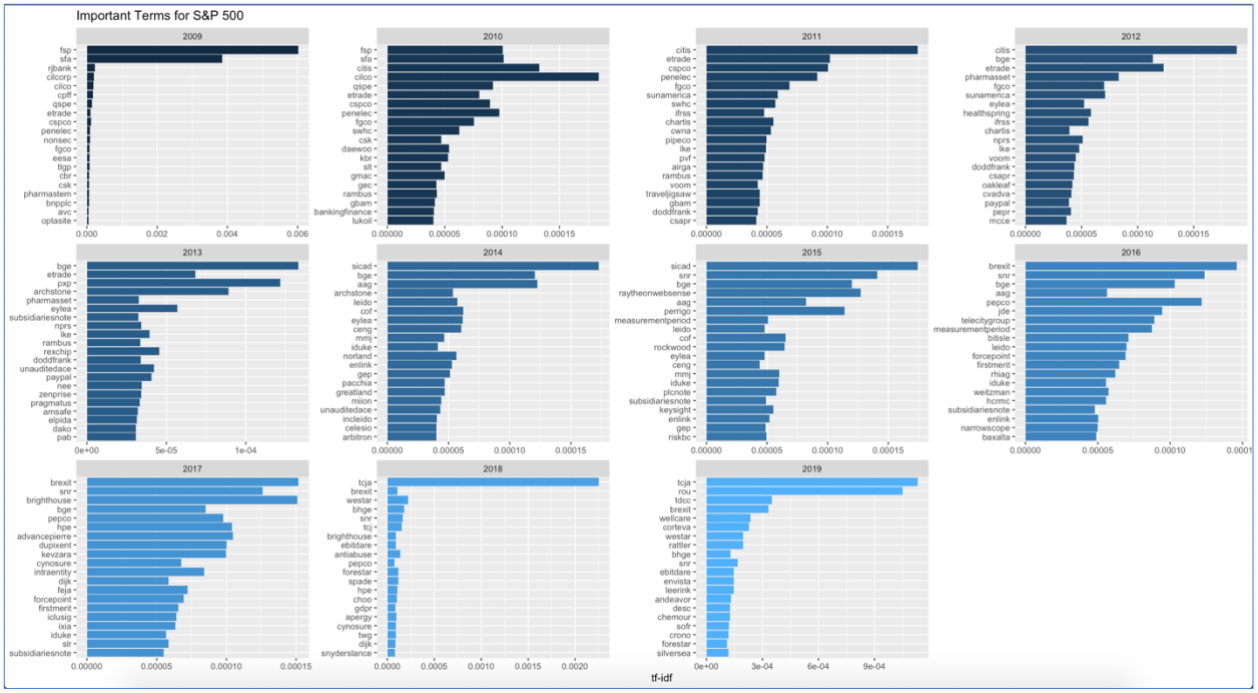
```
  print(paste(i, "of", nrow(market_level), "10Q reports"))

  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}
```

TF-IDF Analysis

```
# ----- TF-IDF Market Level
market_level <- dbGetQuery(conn, 'SELECT year_filed, cleaned_noun FROM master
_index')

market_tokens <- market_level %>%
  unnest_tokens(word, cleaned_noun) %>%
  count(year_filed, word, sort = TRUE) %>%
  ungroup() %>%
  bind_tf_idf(word, year_filed, n)
rm(market_level)

market_tokens %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>% group_by(year_f
iled) %>%
  top_n(20) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill = year_filed)) + geom_col(show.legend = FALSE
) +
  labs(x = NULL, y = "tf-idf", title = "Important Terms for S&P 500") +
  facet_wrap(~year_filed, ncol = 4, scales = "free") + coord_flip()
```

Just by plotting the important words across horizon of years, it can be seen how TF-IDF analysis could capture the 'milestones' of the financial market just by looking at the top terms with the highest TF-IDF weights. For instance;

- **2009** – 2010 FSP (Financial Stability Plan) broad initiative rolled by Obama administration in early 2009 to stabilises the U.S. economy that aims to standardise the banking system and provide capital to unstable lending institutions post-financial crisis (Kenton, 2018)
- **2011** – 2012 CITIS
- **2014** – 2015 SICAD, a Venezuelan central bank initiative to ease access for foreign exchange (Cabrera, 2014).
- **2016** – 2019 Brexit
- **2018** – 2019 TCJA (Tax Cust and Jobs Act 2017)

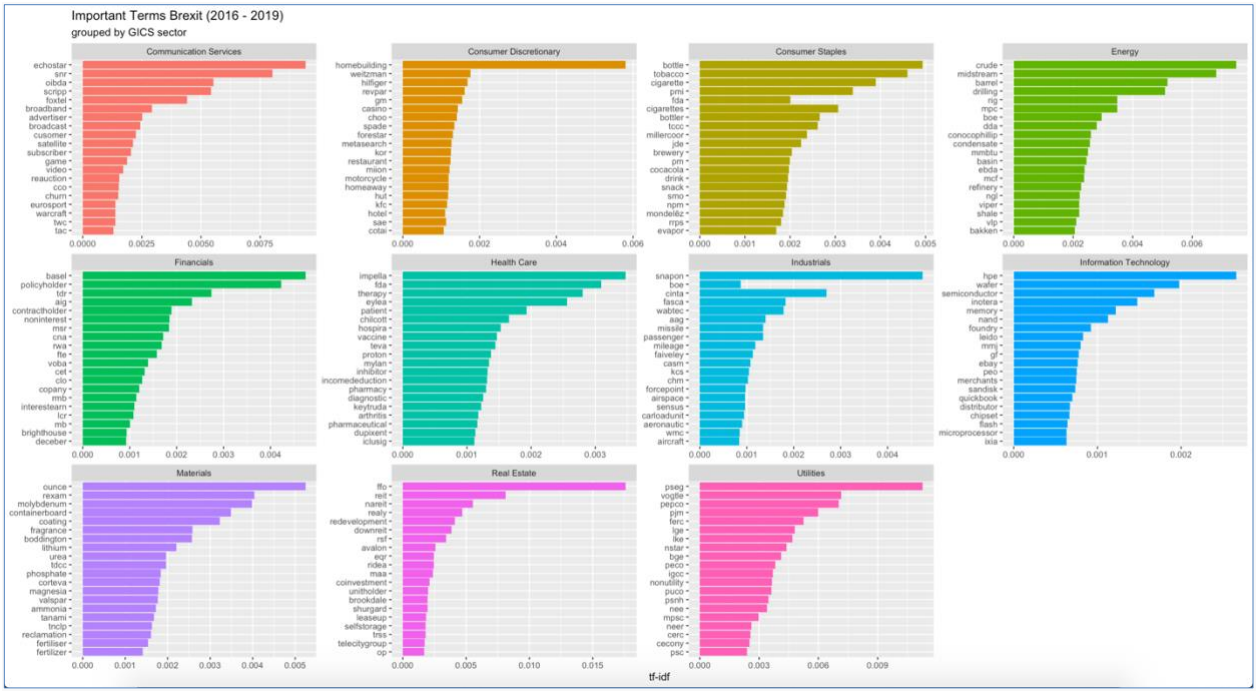



```
# ----- TF-IDF Sector Level


# ----- Define financial Era
financial_era <- c('Post Financial Crisis (2009 - 2010)', 'CITIS (2011 - 2012
)', 'SICAD (2014 - 2015) ', 'Brexit (2016 - 2019)', 'TCJA (2018 - 2019)')
years <- c('2009,2010', '2011,2012', '2014,2015', '2016,2017,2018,2019', '201
8,2019')

for (era in 1:length(financial_era)) {

  gics_level <- dbGetQuery(conn, paste0('SELECT gics_sector, cleaned_noun FRO
M sp500 LEFT JOIN master_index ON master_index.cik =  sp500.cik
                              WHERE year_filed IN (', years[era],')'))

  # ----- Calculate TF-IDF by sector on a financial era
  gics_tokens <- gics_level %>%
    unnest_tokens(word, cleaned_noun) %>%
    count(gics_sector, word, sort = TRUE) %>%
    ungroup() %>%
    bind_tf_idf(word, gics_sector, n)
  rm(gics_level)

  # ----- Plot the important terms
  gics_tokens %>%
    arrange(desc(tf_idf)) %>%
    mutate(word = factor(word, levels = rev(unique(word)))) %>% group_by(gics
```

```
_sector) %>%
    top_n(20) %>%
    ungroup %>%
    ggplot(aes(word, tf_idf, fill = gics_sector)) + geom_col(show.legend = FA
LSE) +
    labs(x = NULL, y = "tf-idf", title = paste("Important Terms", financial_e
ra[era]), subtitle = "grouped by GICS sector") +

    facet_wrap(~gics_sector, ncol = 4, scales = "free") + coord_flip()

}
```

Post financial Crisis

Post Brexit



Likewise, on a sector level in a different era, trends in TF-IDF could capture the progression of humanity. A small instance would be 'Impella', a heart pump device that receives a pre-market approval by the Food and Drug (FDA) administration in the US on 2015, has become the most important word surpassing FDA itself in the Health care industry in the Brexit era (2016 – 2019)

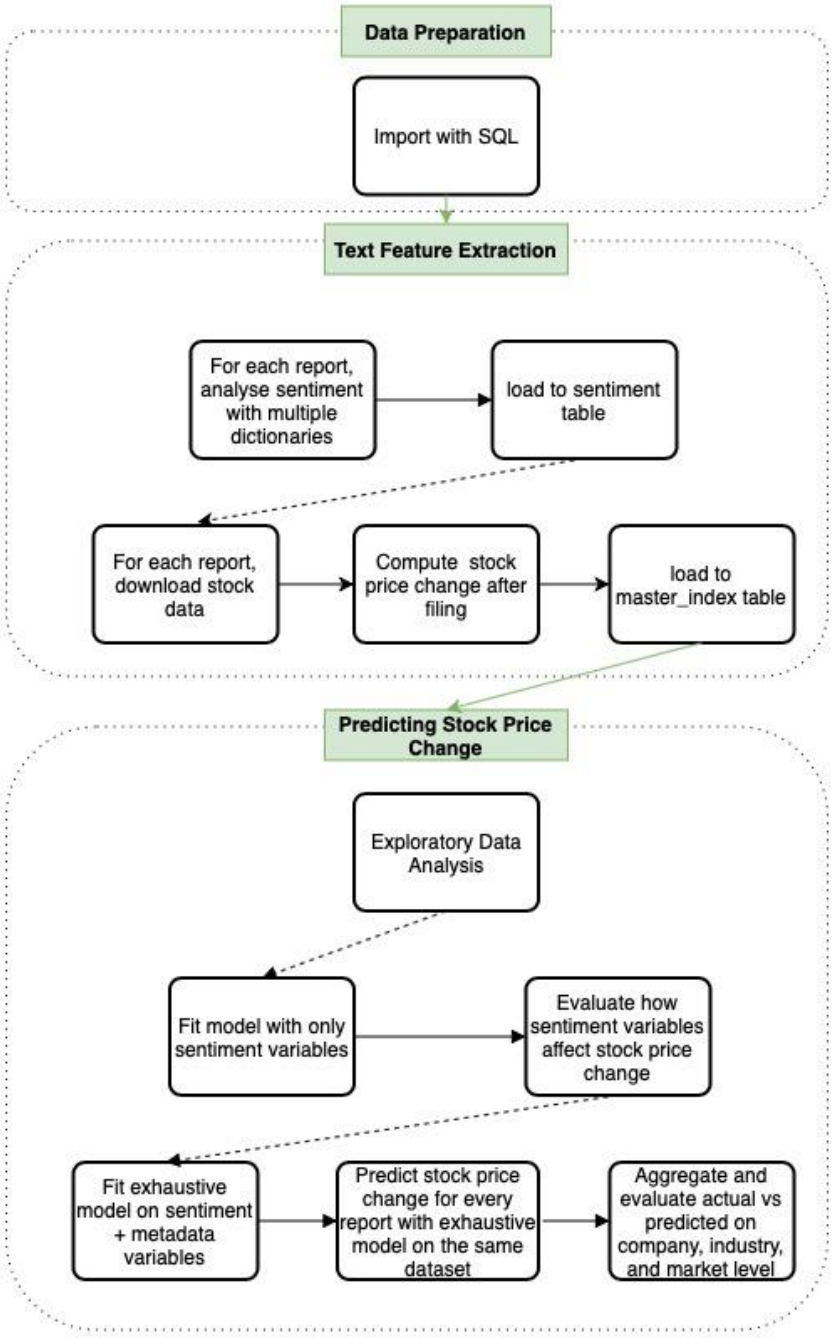
Another example of progression is true for the Information Technology industry where servers has not become a big deal anymore, while semi-conductors and micro-processors might have become more popular in financial reports. Perhaps even in the lunch break conversations between engineers.

Conclusions

Part A has guided this study to build a corpus of 10-K and 10-Q documents and store them in an SQLite database ready to be analysed. Insightful information of major trends in the progression of humanity can also be drawn from the reports of the S&P 500 by utilising TF-IDF measure that gives greater weightage to more important words.

Part B — Sentiment Analysis

Process Workflow



Data Preparation

Importing with SQL will be done as the anlysis furthers.

```
conn <- dbConnect(RSQLite::SQLite(), "edgar.db")
```

Text Feature Extraction

This section aims to quantify the textual information in a report. Sentiment analysis can draw out the subjective information in a text, thus making it a good candidate to quantify financial reports. The underlying assumption is that the financial market will react to how good or bad the financial report describes the state of the company. In an ideal world, it would mean a bad sentiment will lead to a drop in stock price, while a good sentiment will drive the price up.

In addition to the sentiment measures, the change of stock price which will later be used to evaluate predictability is extracted in this section.

Importing Louhran & McDonald's sentiment words

Sentiment word list was also taken from the resource website of Loughran & McDonald.

```
# -- Import LM Dictionary
LM_dictionary_file <- "LoughranMcDonald_SentimentWordLists_2018.xlsx"
sentiment <- c("negative", "positive", "uncertainty", "litigious", "strong_mo
dal","weak_modal", "constraining")

LM_dictionary <- data.frame()
for(s in 1:7) {
  local_df <- tibble(word = tolower(read_excel(LM_dictionary_file, sheet = s+
1)$word), sentiment = sentiment[s] )
  LM_dictionary <- bind_rows(LM_dictionary, local_df)
}

rm(local_df)

dummy_LM <- tibble(accession_number = 'dummy', positive = 0, negative = 0, un
certainty = 0, litigious = 0, constraining = 0, strong_modal = 0, weak_modal
= 0)
dummy_nrc <- tibble(accession_number = 'dummy', positive = 0, negative = 0, a
nger = 0, fear = 0, trust = 0, sadness = 0, surprise = 0, disgust = 0, joy =
0, anticipation = 0)
dummy_bing <- tibble(accession_number = 'dummy', positive = 0, negative = 0)
```

Calculating Sentiment Measures and Loading to sentiment table

Sentiments are calculated using different dictionaries including Syuzhet, Vader, Loughran & McDonald, NRC, Sentimentr, Bing, and Afinn.

```
cik <- dbGetQuery(conn, 'SELECT distinct(cik) as cik FROM master_index')$cik


for (c in 1:length(cik)) {
  df_cik <- dbGetQuery(conn, paste0('SELECT accession_number, cleaned_text FR
OM master_index WHERE cik = "',cik[c], '"'))

  for (r in 1:nrow(df_cik)) {

  df_report <- dbGetQuery(conn, paste0('SELECT accession_number, cleaned_text
FROM master_index WHERE accession_number = "',df_cik$accession_number[r], '"'
))
  # ----- Sentiment Syuzhet, Vader, and n_words
  sentiment_syuzhet_vader <- textfeatures(df_report$cleaned_text, normalize =
FALSE, word_dims = FALSE, sentiment = TRUE) %>%
    select(sent_syuzhet,sent_vader) %>%
    mutate(accession_number = df_report$accession_number)

  # ----- Text Complexity
  tokenised <- df_report %>%
    unnest_tokens(word, cleaned_text)

  n_words <- tokenised %>%
    group_by(accession_number) %>%
    count(accession_number)

  n_complex <- tokenised %>%
    group_by(word, accession_number) %>%
    mutate(complexity = nchar( gsub( "[^X]", "", gsub( "[aeiouy]+", "X", tolo
wer( word ))))) %>%
    filter(complexity >=3) %>%
    group_by(accession_number) %>%
    count(accession_number)

  complexity <- tibble(accession_number = n_words$accession_number,
                       complexity = n_complex$n / n_words$n)

  rm(n_complex)

  # ----- Sentiment LoughranMcDonald
  tokens_LM <- tokenised %>%
    inner_join(LM_dictionary)

  word_count_LM <- tokens_LM %>% group_by(accession_number) %>% summarise(LM_
```

```
total_words =n())

  sentiment_LM <- tokens_LM %>%
    group_by(accession_number,sentiment) %>%
    summarise(total_sentiment = n()) %>%
    spread(sentiment, total_sentiment, fill = 0) %>%
    bind_rows(dummy_LM) %>%
    left_join(word_count_LM) %>%
    mutate(LM_sent = positive - negative,
           LM_positive = positive / LM_total_words,
           LM_negative = negative / LM_total_words,
           LM_uncertainty = uncertainty / LM_total_words,
           LM_litigious = litigious / LM_total_words,
           LM_constraining = constraining / LM_total_words,
           LM_strong_modal = strong_modal / LM_total_words,
           LM_weak_modal = weak_modal / LM_total_words) %>%
    select(-c(positive, negative, uncertainty, litigious, constraining, stron
g_modal, weak_modal)) %>%
    filter(accession_number != 'dummy')

  rm(tokens_LM, word_count_LM)

  # ----- Sentimentr
  text <- get_sentences(df_report$cleaned_text)
  sentimentr <- tibble(accession_number = df_report$accession_number,
                       sentimentr = as.data.frame(sentiment_by(text))$ave_sen
timent)

  rm(text)

  # ----- Sentiment Afinn
  sentiment_afinn <- tokenised %>%
    inner_join(get_sentiments("afinn")) %>%
    group_by(accession_number) %>%
    summarise(afinn_sent = sum(value))

  # ----- Sentiment bing
  tokens_bing <- tokenised %>%
    inner_join(get_sentiments("bing"))

  word_count_bing <- tokens_bing %>% group_by(accession_number) %>% summarise
(bing_total_words =n())

  sentiment_bing <- tokens_bing %>%
    group_by(accession_number,sentiment) %>%
     summarise(total_sentiment = n()) %>%
    spread(sentiment, total_sentiment, fill = 0) %>%
    bind_rows(dummy_bing) %>%
    left_join(word_count_bing) %>%
```

```
    mutate(bing_sent = positive - negative,
           bing_positive = positive/bing_total_words,
           bing_negative = negative/bing_total_words) %>%
    select(-c(positive, negative)) %>%
    filter(accession_number != 'dummy')

    rm(tokens_bing, word_count_bing)

  # ----- Sentiment NRC
  tokens_nrc <- tokenised %>%
    inner_join(get_sentiments("nrc"))

  word_count_nrc <- tokens_nrc %>% group_by(accession_number) %>% summarise(n
rc_total_words =n())

  sentiment_nrc <- tokens_nrc %>%
    group_by(accession_number,sentiment) %>%
    summarise(total_sentiment = n()) %>%
    spread(sentiment, total_sentiment, fill = 0) %>%
    bind_rows(dummy_nrc) %>%
    left_join(word_count_nrc) %>%
    mutate(nrc_sent = positive - negative,
           nrc_positive = positive / nrc_total_words,
           nrc_negative = negative / nrc_total_words,
           nrc_anger = anger/nrc_total_words,
           nrc_fear = fear/nrc_total_words,
           nrc_trust = trust/nrc_total_words,
           nrc_sadness = sadness/nrc_total_words,
           nrc_surprise = surprise/nrc_total_words,
           nrc_disgust = disgust/nrc_total_words,
           nrc_joy = joy/nrc_total_words,
           nrc_anticipation = anticipation/nrc_total_words) %>%
    select(-c(positive, negative, anger, trust, sadness, surprise, disgust,jo
y, anticipation, fear )) %>%
    filter(accession_number != 'dummy')

  rm(tokens_nrc, word_count_nrc)

  # ----- Merging Sentiment Features
  sentiment_df <- sentiment_LM %>%
    left_join(complexity) %>%
    left_join(sentimentr) %>%
    left_join(sentiment_afinn) %>%
    left_join(sentiment_bing) %>%
    left_join(sentiment_nrc) %>%
    left_join(sentiment_syuzhet_vader)

  rm(sentiment_LM, complexity, sentimentr, sentiment_afinn, sentiment_bing, s
entiment_nrc, sentiment_syuzhet_vader, n_words, tokenised)
```

```
  # ----- Insertion to SQL table
  dbWriteTable(conn,"sentiment", sentiment_df, append = TRUE) # insert to sen
timent_df Table

  rm(sentiment_df)

  }

  print(paste(c, "of", length(cik), "sentiment calculated"))
}

rm(df_cik, df_report, dummy_bing, dummy_LM, dummy_nrc, LM_dictionary, cik, c,
r, sentiment, LM_dictionary_file)
```

Downloading Stock Price

An underlying assumption for this section is that the effect of the filing public release can be captured by comparing price Day-7 with Day+3 of the filing date.

In this section, two measures of stock price change are calculated; the log return adjusted value, where the log return difference between 2 dates are taken, while the adjusted price change takes the ratio between the new adjusted stock price (price after posting a dividend) is divided by the old adjusted price – 100%, where 0% means no price change and 10% means a 10% increase in stock price. Later in the exploratory data ananalysis section one of the measures will be used to guide this study.

```
# ----- Set up columns
dbExecute (conn, 'ALTER TABLE master_index ADD COLUMN return_adjusted_price d
ouble;')
dbExecute (conn, 'ALTER TABLE master_index ADD COLUMN price_adjusted_ratio do
uble;')
```

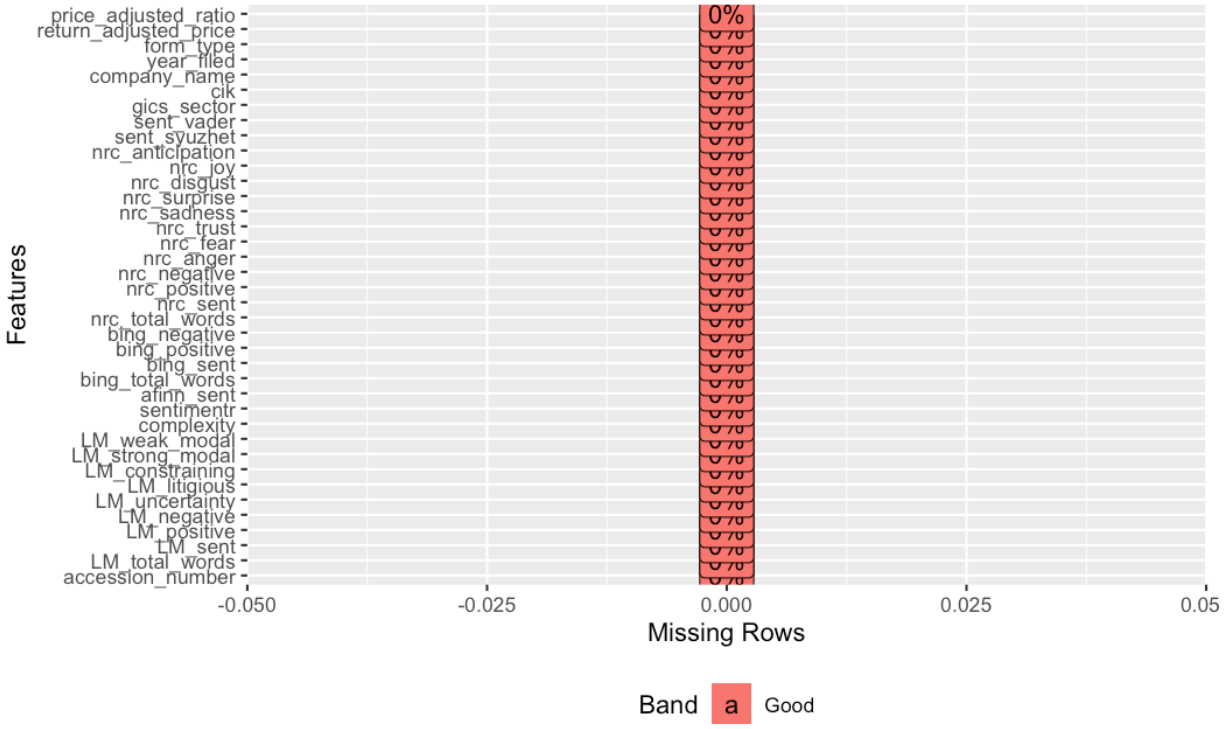
```
# ----- Downloading and Loading to master_index table
cik <- dbGetQuery(conn, 'SELECT distinct(cik) as cik FROM master_index')$cik

for (c in 1:length(cik)) {

  # ----- Import per CIK
  df_cik <- dbGetQuery(conn, paste0('SELECT master_index.cik, form_type, date
_filed, accession_number, symbol FROM master_index
                            LEFT JOIN (SELECT cik, symbol FROM sp500 group by
cik) AS sp500
                            ON master_index.cik = sp500.cik
                            WHERE master_index.cik = "',  cik[c]   ,'"')) %>%
mutate(date_filed = as.Date(date_filed, origin="1970-01-01"))
```

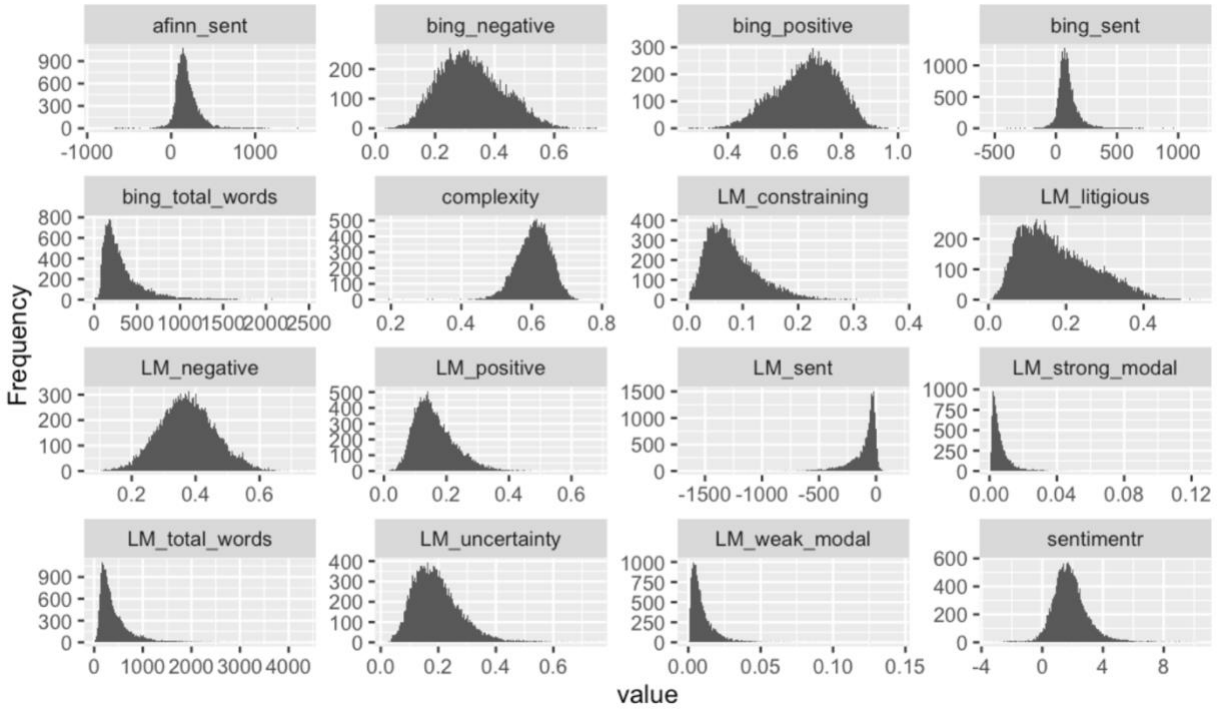
```r
  for (r in 1:nrow(df_cik)) {
    tryCatch({
    df_report <- df_cik[r,] # iterate for every row

    # ----- Get stock information
    stock_data <- BatchGetSymbols(tickers = df_report$symbol,
                                  first.date= df_report$date_filed - 7,
                                  last.date= df_report$date_filed + 3,
                                  type.return="log")

    # ----- Filter the 2nd day and the last day
    stock_data_filtered <- stock_data[[2]] %>%
      filter(ref.date == max(ref.date) | row_number() == 2) %>%
      arrange(desc(ref.date))

    # ----- Calculate stock price change on log scale
    return_adjusted_price <- stock_data_filtered$ret.closing.prices[1] - stoc
k_data_filtered$ret.closing.prices[2] # return difference
    price_adjusted_ratio <- (stock_data_filtered$price.adjusted[1] / stock_da
ta_filtered$price.adjusted[2]) - 1 # stock price ratio

    accession_number <- df_report$accession_number[1]

    # ----- Update to DB
    dbExecute(conn, paste0("UPDATE master_index SET return_adjusted_price = "
,return_adjusted_price ,", price_adjusted_ratio = ",price_adjusted_ratio ,"
                           WHERE accession_number = '",accession_number ,"
'"))


    print(paste(accession_number, "has been processed..."))

  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
  }

  print(paste(c, "of", length(cik), "stock price change added"))
}
```

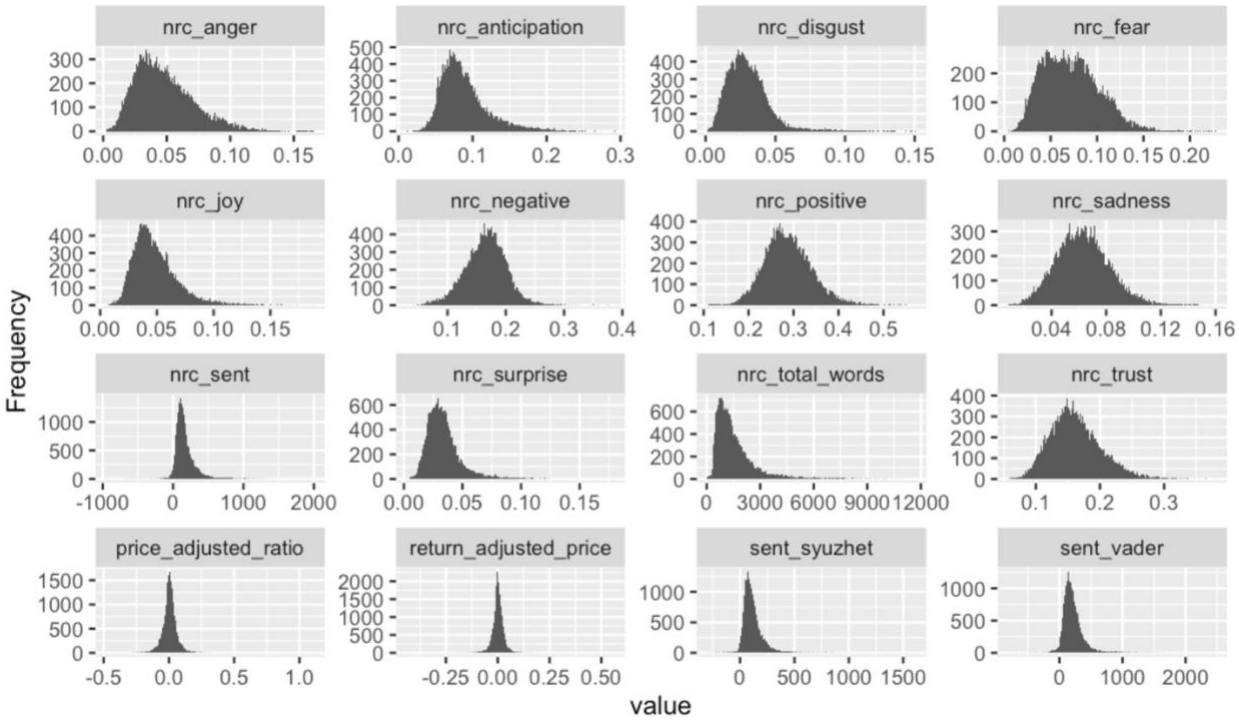
Predicting Stock Price Change

Exploratory Data Analysis

```
# ----- Data Prep
sentiment_data <- dbGetQuery(conn, 'SELECT sentiment.*, gics_sector, master_i
ndex.cik, company_name, year_filed, form_type, return_adjusted_price, price_a
djusted_ratio FROM sentiment
                             INNER JOIN master_index ON sentiment.accession_n
umber = master_index.accession_number
                             INNER JOIN (SELECT * FROM sp500 group by cik) AS
sp500 ON master_index.cik = sp500.cik
                             ') %>% mutate(year_filed = as.factor(year_filed)
,
                                                                    company_n
ame = as.factor(company_name),
                                                                    cik = as.
factor(cik),
                                                                    gics_sect
or = as.factor(gics_sector),
                                                                    accession
_number = as.factor(accession_number),
                                                                    price_adj
usted_ratio = price_adjusted_ratio*100)



load('temp/sentiment_data.rda')



sentiment_data <- sentiment_data %>%
  filter(LM_total_words > 10 & bing_total_words > 10, nrc_total_words > 10) %
>%
  filter(price_adjusted_ratio != !is.na(price_adjusted_ratio) | return_adjust
ed_price != is.na(return_adjusted_price)) %>%
      mutate_if(is.numeric, funs(ifelse(is.na(.), 0, .)))

save(sentiment_data, file = '/Volumes/Buku Gibran/edgar/temp/sentiment_data.r
da')

plot_missing(sentiment_data)
plot_histogram(sentiment_data)
```

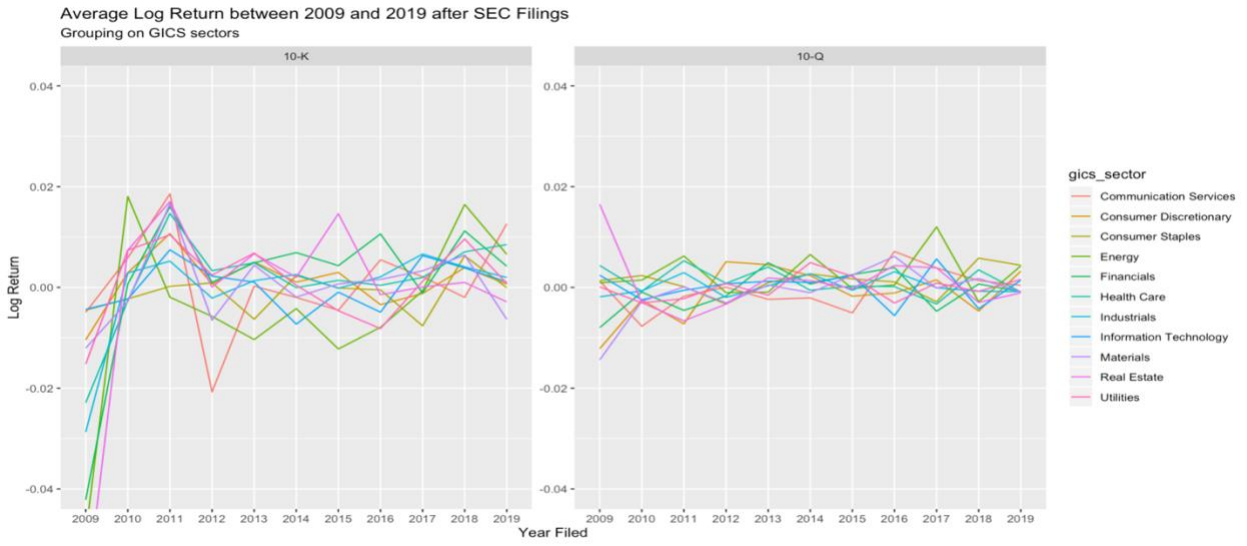

price_adjusted_ratio
return_adjusted_price
form_type
year_filed
company_name
cik
gics_sector
sent_vader
sent_syuzhet
nrc_anticipation
nrc_joy
nrc_disgust
nrc_surprise
nrc_sadness
nrc_trust
nrc_fear
nrc_anger
nrc_negative
nrc_positive
nrc_sent
nrc_total_words
bing_negative
bing_positive
bing_sent
bing_total_words
afinn_sent
sentimentr
complexity
LM_weak_modal
LM_strong_modal
LM_constraining
LM_litigious
LM_uncertainty
LM_negative
LM_positive
LM_sent
LM_total_words
accession_number
0%
Features
-0.050
-0.025
0.000
0.025
0.05
Missing Rows
Band
a
Good

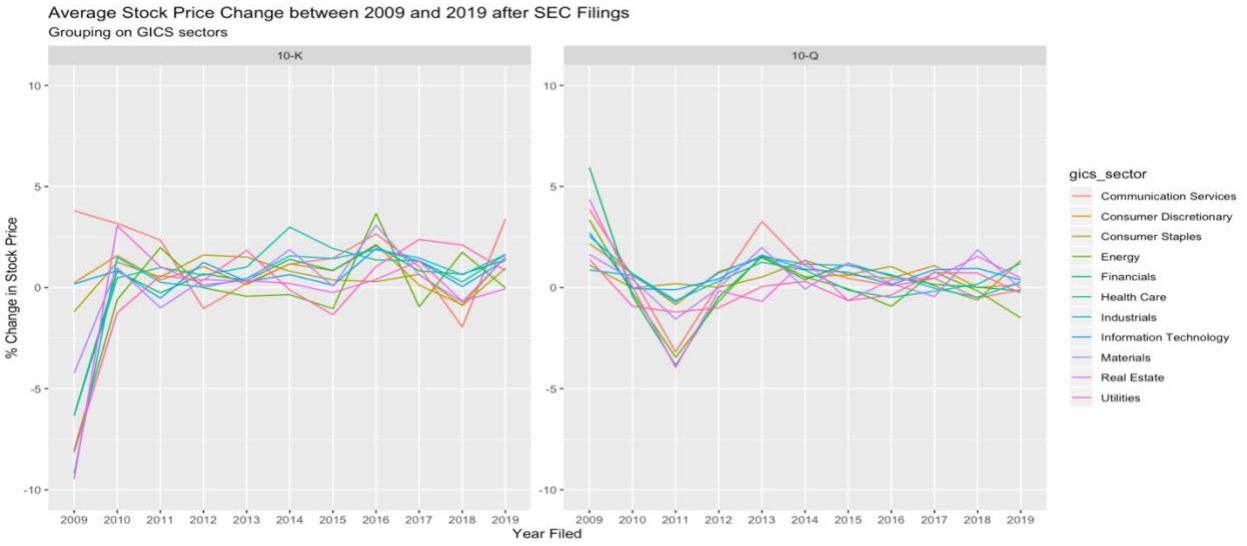

afinn_sent
bing_negative
bing_positive
bing_sent
bing_total_words
complexity
LM_constraining
LM_litigious
LM_negative
LM_positive
LM_sent
LM_strong_modal
LM_total_words
LM_uncertainty
LM_weak_modal
sentimentr
Frequency
value




```r
# ----- Plot Average Return
return_avg <- sentiment_data %>%
  group_by(gics_sector, year_filed, form_type) %>%
  summarise(return_adjusted_price = mean(return_adjusted_price))

return_avg %>%
  ggplot(aes(x = year_filed, y = return_adjusted_price, color = gics_sector))
+
  geom_line(aes(group = gics_sector))  +
  labs(title = 'Average Log Return between 2009 and 2019 after SEC Filings',
subtitle='Grouping on GICS sectors', x = 'Year Filed', y = 'Log Return', lege
nd = 'GICS Sector') + coord_cartesian(ylim=c(-0.04,0.04)) +
  facet_wrap(~form_type, ncol = 4, scales = "free")


# ----- Plot Average Stock Price Change
price_avg <- sentiment_data %>%
  group_by(gics_sector, year_filed, form_type) %>%
  summarise(price_adjusted_ratio = mean(price_adjusted_ratio))

price_avg %>%
  ggplot(aes(x = year_filed, y = price_adjusted_ratio, color = gics_sector))
+
  geom_line(aes(group = gics_sector))  +
  labs(title = 'Average Stock Price Change between 2009 and 2019 after SEC Fi
```

```
lings', subtitle='Grouping on GICS sectors', x = 'Year Filed', y = '% Change
in Stock Price', legend = 'GICS Sector') + coord_cartesian(ylim=c(-10,10)) +
  facet_wrap(~form_type, ncol = 4, scales = "free")

rm(return_avg,price_avg)
```





The average stock price change where the % change is used seems to have lower variance than the average log return. Additionally, the % change is easier to interpret and has been normalised as it is already in a percentage format. Moreover, it seems to confirm the known trend of post-financial crisis in 2009. Therefore, author decided to conduct analysis uing the stock price % change instead of the log return.

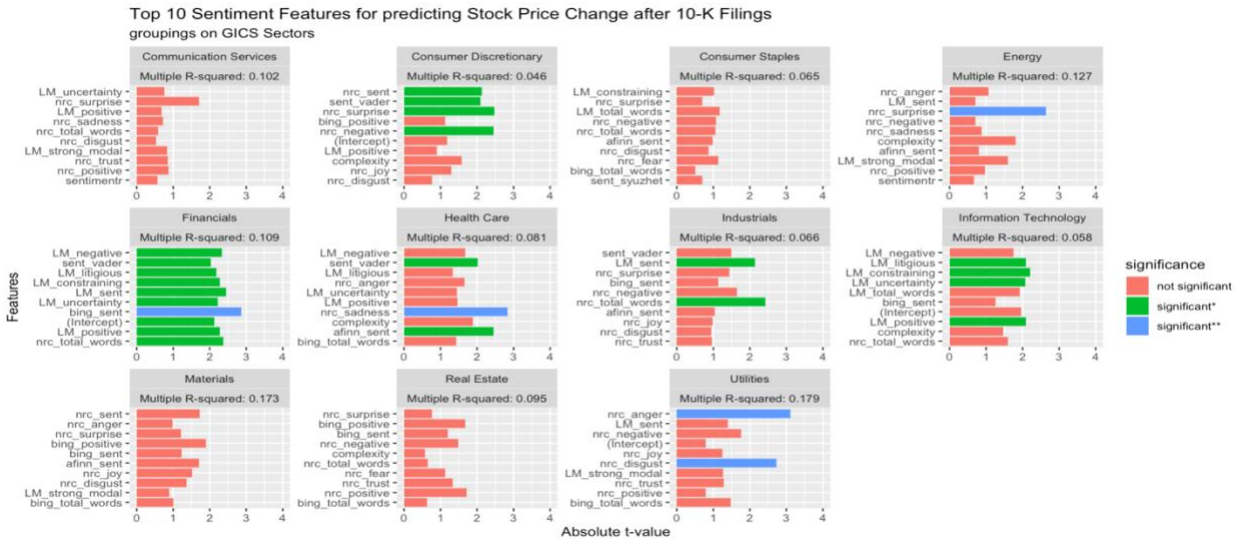
How Sentiment Analysis affects Stock Price Change

```
# ----- Individual Dictionaries
load('/Volumes/Buku Gibran/edgar/temp/sentiment_data.rda')


sentiment_regression_data <-sentiment_data %>% filter(form_type ==  '10-K') %
>% select(-c(form_type, year_filed, cik, company_name, return_adjusted_price,
gics_sector, accession_number))x
str(sentiment_regression_data)

reg_10k_LM <- lm(price_adjusted_ratio ~ LM_total_words + LM_sent + LM_positiv
e + LM_negative + LM_uncertainty + LM_litigious + LM_constraining + LM_strong
_modal + LM_weak_modal + complexity, data = sentiment_regression_data)
reg_10k_bing <- lm(price_adjusted_ratio ~ bing_total_words + bing_sent + bing
_positive + bing_negative, data = sentiment_regression_data)
reg_10k_sentimentr <- lm(price_adjusted_ratio ~ sentimentr, data = sentiment_
regression_data)
reg_10k_afinn <- lm(price_adjusted_ratio ~ afinn_sent, data = sentiment_regre
ssion_data)
reg_10k_nrc <- lm(price_adjusted_ratio ~ nrc_sent + nrc_total_words + nrc_pos
itive + nrc_negative + nrc_anger + nrc_fear + nrc_trust + nrc_sadness + nrc_s
urprise + nrc_disgust + nrc_joy + nrc_anticipation, data = sentiment_regressi
on_data)
reg_10k_syuzhet <- lm(price_adjusted_ratio ~ sent_syuzhet, data = sentiment_r
egression_data)
reg_10k_vader <- lm(price_adjusted_ratio ~ sent_vader, data = sentiment_regre
ssion_data)

library(stargazer)

stargazer::stargazer(reg_10k_LM,reg_10k_bing,reg_10k_sentimentr, reg_10k_afin
n, reg_10k_nrc, reg_10k_syuzhet, reg_10k_vader, type = "text")

sector <- unique(sentiment_data$gics_sector)
```

Out of all dictionaries examined, Loughran & McDonald's has the biggest fit of 0.06, followed by NRC with 0.04, and ending with Syuzhet at the buttom of the list at 0.0004 coefficient of determination (R-squared).

	Dependent variable:						
	price_adjusted_ratio						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
LM_total_words	-0.001						
	(0.001)						
LM_sent	0.001						
	(0.003)						
LM_positive	-6.625						
	(6.300)						
LM_negative	-7.933						
	(6.079)						
LM_uncertainty	-8.686						
	(6.240)						
LM_litigious	-5.560						
	(6.285)						
LM_constraining	-7.051						
	(6.199)						
LM_strong_modal	-9.619						
	(12.439)						
LM_weak_modal							
complexity	3.059**						
	(1.362)						
bing_total_words		-0.00005					
		(0.001)					
bing_sent		-0.004					
		(0.002)					
bing_positive		2.973**					
		(1.304)					
bing_negative							
sentimentr			0.121				
			(0.073)				
afinn_sent				-0.0001			
				(0.001)			
nrc_sent					0.001		
					(0.001)		
nrc_total_words					-0.0003**		
					(0.0001)		
nrc_positive					-1.860		
					(3.200)		
nrc_negative					-0.216		
					(3.041)		
nrc_anger					5.022		
					(5.584)		
nrc_fear					-0.398		
					(3.484)		
nrc_trust					2.484		
					(2.538)		
nrc_sadness					5.202		
					(4.538)		
nrc_surprise					-8.496		
					(5.751)		
nrc_disgust					-5.955		
					(5.465)		
nrc_joy					1.692		
					(4.715)		
nrc_anticipation							
sent_syuzhet						0.0003	
						(0.001)	
sent_vader							-0.0002
							(0.001)
Constant	6.373	-1.343	0.244*	0.442***	0.773	0.404***	0.470***
	(6.036)	(0.910)	(0.140)	(0.119)	(2.059)	(0.126)	(0.119)
Observations	4,240	4,240	4,240	4,240	4,240	4,240	4,240
R2	0.006	0.003	0.001	0.00000	0.004	0.00002	0.00004
Adjusted R2	0.004	0.002	0.0004	-0.0002	0.001	-0.0002	-0.0002
Residual Std. Error	5.100 (df = 4230)	5.103 (df = 4236)	5.108 (df = 4238)	5.110 (df = 4238)	5.106 (df = 4228)	5.110 (df = 4238)	5.110 (df = 4238)
F Statistic	2.837*** (df = 9; 4230)	4.348*** (df = 3; 4236)	2.706 (df = 1; 4238)	0.008 (df = 1; 4238)	1.483 (df = 11; 4228)	0.092 (df = 1; 4238)	0.156 (df = 1; 4238)
Note:							*p<0.1; **p<0.05; ***p<0.01

```r
# ----- Model sentiment for every Sector
model_output_sector <- data.frame()

for (s in 1:length(sector)) {
  model_data <- sentiment_data %>% filter(form_type == "10-K", gics_sector == sector[s]) %>%
  select(-c(accession_number,year_filed, form_type, cik, company_name, return_adjusted_price, gics_sector))

  # ----- Modelling
  model <- lm(price_adjusted_ratio ~., data = model_data)

  # ----- Prep data for plotting
  local_df <- head(as.data.frame(summary(model)$coefficients) %>%
    tibble::rownames_to_column() %>%
    mutate(absolute_t_value = abs(`t value`)) %>%
    arrange(desc(absolute_t_value)) , 10) %>%
    mutate(rowname=factor(rowname, levels=rowname)) %>%
    mutate(significance = case_when(`Pr(>|t|)` <= 0.001 ~ 'significant***', `Pr(>|t|)` <= 0.01 ~ 'significant**', `Pr(>|t|)` <= 0.05 ~ 'significant*', TRUE ~ 'not significant')) %>%
    mutate(gics_sector = sector[s]) %>%
    mutate(r_squared = paste('Multiple R-squared:',as.character(round(summary(model)$r.squared,3))))

  model_output_sector <- bind_rows(model_output_sector, local_df)
}

ggplot(model_output_sector, aes(x = reorder(rowname, absolute_t_value), y = absolute_t_value, fill=significance)) + geom_bar(stat = "identity") +
  labs(title ='Top 10 Sentiment Features for predicting Stock Price Change after 10-K Filings', subtitle = 'groupings on GICS Sectors',y = 'Absolute t-value', x = 'Features') +
  ylim(0,4) +  coord_flip() +
  facet_wrap(~gics_sector+r_squared, ncol = 4, scales = "free")

rm(model_output_sector)
```



Different industries have different sentiment variable significance and fit. For instance, the stock price change of Financials industry could be majorly explained by Loughran-McDonald with a 0.109 predictability, while in Utilities industry NRC tend to explain better.

Modelling 10-K: Building Exhaustive Model to predict Stock Price Change

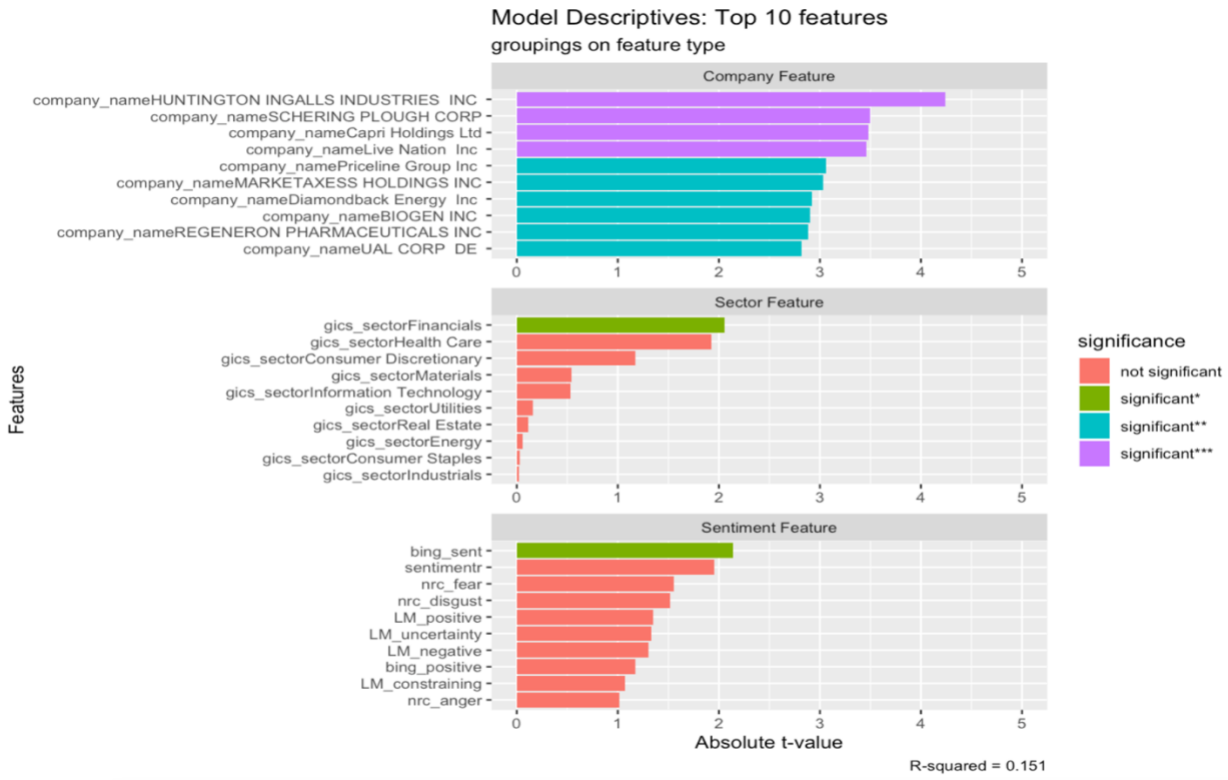
The idea of an exhaustive model is to use all sentiment variables available in addition to the meta-data of the report such as company identifier and GICS sector as model features to predict *stock price % change of a report*.

Since the aim of an exhaustive model is for the model to be able to predict future stock price change, the year of the report filed is not used for model fitting as it is generally a time-related variable, not a category one like sectors. This boils down to the fact that when a linear model is fitted using a categorical variable it will assign a coefficient to each of its levels. However, for future data, it would be the case that the model would not recoginse the year since it has not been pre-trained before. Therefore, author did not include year filed as input variable for model fitting.

```
load(file = '/Volumes/Buku Gibran/edgar/temp/sentiment_data.rda')
# ----- Stock Price Modelling
model_data_10k <- sentiment_data %>%
  filter(form_type == "10-K") %>%
select(-c(form_type, cik, return_adjusted_price))

accession_number <- model_data_10k$accession_number
year_filed <- model_data_10k$year_filed
model_data_10k <- model_data_10k %>% select(-c(year_filed, accession_number))
# year_filed is removed so it is not considered by the model
```

```r
model_10k <- lm(price_adjusted_ratio ~., data = model_data_10k, na.action=na.exclude)

save(model_10k, file = '/Volumes/Buku Gibran/edgar/temp/model_10k.rda')

load(file = '/Volumes/Buku Gibran/edgar/temp/model_10k.rda')
model_data_10k$predicted_ratio <- stats::predict(model_10k, newdata = model_data_10k %>% select(-c(price_adjusted_ratio)))
model_data_10k$year_filed <- year_filed # add back year_filed as to help grouping
model_data_10k$accession_number <- accession_number

save(model_data_10k, file = '/Volumes/Buku Gibran/edgar/temp/model_data_10k.rda')


local_df <- head(as.data.frame(summary(model_10k)$coefficients) %>%
    tibble::rownames_to_column() %>%
    mutate(absolute_t_value = abs(`t value`)) %>%
    arrange(desc(absolute_t_value)) , 1000) %>%
    mutate(rowname=factor(rowname, levels=rowname)) %>%
    mutate(significance = case_when(`Pr(>|t|)` <= 0.001 ~ 'significant***', `Pr(>|t|)` <= 0.01 ~ 'significant**', `Pr(>|t|)` <= 0.05 ~ 'significant*', TRUE ~ 'not significant')) %>%
    mutate(r_squared = paste('Multiple R-squared:',as.character(round(summary(model_10k)$r.squared,3)))) %>%
    mutate(category = case_when(grepl("company_name", rowname, fixed = TRUE) ~ 'Company Feature', grepl("gics_sector", rowname, fixed = TRUE) ~ 'Sector Feature', TRUE ~ 'Sentiment Feature'))


feature_category <- c('Sentiment Feature', 'Sector Feature', 'Company Feature')

model_df <- data.frame()
for(v in 1:length(feature_category)) {
  top_n <- local_df %>% filter(category == feature_category[v]) %>% arrange(desc(absolute_t_value))
  top_n <- top_n[1:10,]
  model_df <- bind_rows(model_df, top_n)
}


ggplot(model_df, aes(x = reorder(rowname, absolute_t_value), y = absolute_t_value, fill=significance)) + geom_bar(stat = "identity") +
  labs(title ='Model Descriptives: Top 10 features', subtitle = 'groupings on feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squared = ', as.character(round(summary(model_10k)$r.squared,3)) )) +
```

```
  ylim(0,5) +  coord_flip() +
  facet_wrap(~category, nrow = 4, scales = "free")


rm(model_df, local_df, top_n)
```



Despite the fact that Bing sentiment did not performed any better than LM sentiment or NRC individually, when piled up with other meta-data features, only Bing sentiment shows significant effect to stock price change. It is most likely because the effect explained by LM or NRC has been absorbed by one of company or sector features. All features together can explain a variance of 0.151.

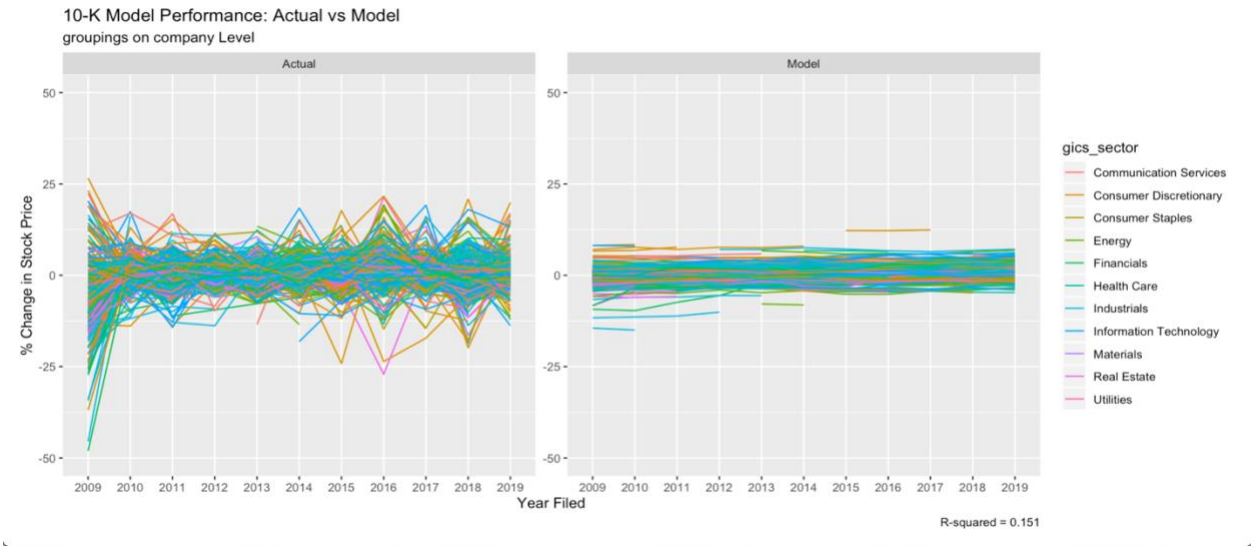
Model Evaluation: 10-K

As mentioned earlier, the exhaustive model is fitted on a report level not in an aggregated state. Garrett (2003) stated that sign and significance of coefficient estimates between unaggregated and aggregated data can differ, which often cause misleading conclusions.

However, it is still within our interest to evaluate the model on different aggregation levels such as company, sector, and market level. Therefore, instead of fitting multiple models for different aggregations, author utilises a machine learning like approach by predicting the stock price change for every report first. The catch is instead of using different training-test sets, the dataset to fit the model is the same as the one used to predict using the predict

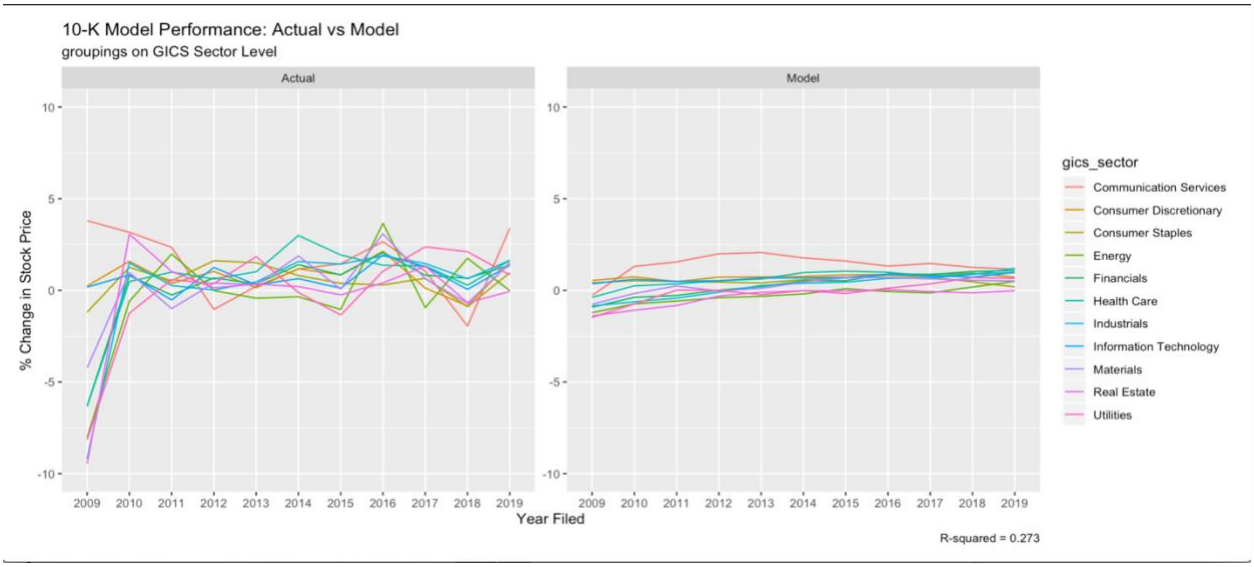
function. Finally, the actual stock price change and the predicted stock price change are aggregated and compared using the R-squared measure on different aggregation levels.

```
load(file = '/Volumes/Buku Gibran/edgar/temp/model_data_10k.rda')

rsq <- function (x, y) cor(x, y) ^ 2 # setup R-squared calculation

# ----- Company Level
company_agg <-  model_data_10k %>%
     group_by(year_filed, company_name, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

company_agg_rsquared <- round(rsq(company_agg$actual_ratio, company_agg$predi
cted_ratio), 3) # calculate R-squared between actual and predicted

company_agg_actual <- company_agg %>% select(-predicted_ratio) %>% mutate(pri
ce_adjusted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
company_agg_predicted <- company_agg %>% select(-actual_ratio) %>% mutate(pri
ce_adjusted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_r
atio)

company_agg <- bind_rows(company_agg_actual, company_agg_predicted)

# ----- Plot Actual vs Model
ggplot(company_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gic
s_sector)) + geom_line(aes(group = company_name)) + coord_cartesian(ylim=c(-5
0,50)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-K Mo
del Performance: Actual vs Model', subtitle = 'groupings on company Level',y
= '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared =
', company_agg_rsquared))

rm(company_agg, company_agg_rsquared, company_agg_actual, company_agg_predict
ed)
```



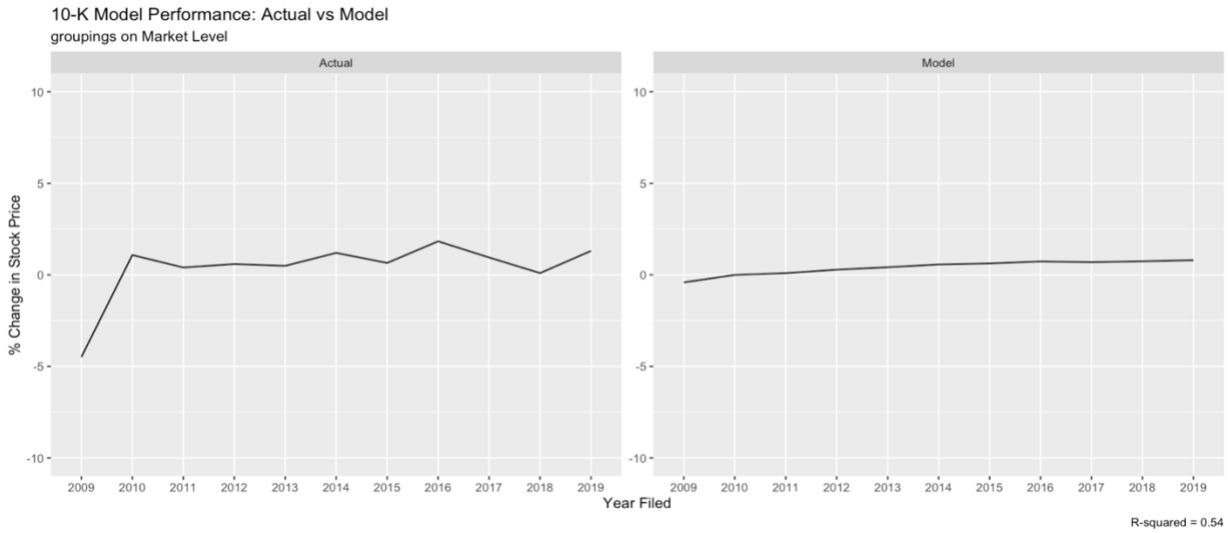
At a company level each line represents one S&P company. The model can explain a 0.151 variance.

```
# ----- GICS Industry Level
gics_agg <-  model_data_10k %>%
     group_by(year_filed, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

gics_agg_rsquared <- round(rsq(gics_agg$actual_ratio, gics_agg$predicted_rati
o), 3) # calculate R-squared between actual and predicted

gics_agg_actual <- gics_agg %>% select(-predicted_ratio) %>% mutate(price_adj
usted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
gics_agg_predicted <- gics_agg %>% select(-actual_ratio) %>% mutate(price_adj
usted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_ratio)

gics_agg <- bind_rows(gics_agg_actual, gics_agg_predicted)

# ----- Plot Actual vs Model
ggplot(gics_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gics_s
ector)) + geom_line(aes(group = gics_sector)) + coord_cartesian(ylim=c(-10,10
)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-K Model
Performance: Actual vs Model', subtitle = 'groupings on GICS Sector Level',y
= '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared =
', gics_agg_rsquared))

rm(gics_agg, gics_agg_rsquared, gics_agg_actual, gics_agg_predicted)
```



As the data is aggregated to a sector level, the R-squared value increases to 0.273. This is expected as companies in the the same sector could off-set or average out each other's performance, leading to a more predictable trend.

```
# ----- Market Level
market_agg <-  model_data_10k %>%
     group_by(year_filed) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

market_agg_rsquared <- round(rsq(market_agg$actual_ratio, market_agg$predicte
d_ratio), 3) # calculate R-squared between actual and predicted

market_agg_actual <- market_agg %>% select(-predicted_ratio) %>% mutate(price
_adjusted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
market_agg_predicted <- market_agg %>% select(-actual_ratio) %>% mutate(price
_adjusted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_rat
io)

market_agg <- bind_rows(market_agg_actual, market_agg_predicted)

# ----- Plot Actual vs Model
ggplot(market_agg, aes(x = year_filed, y = price_adjusted_ratio, group =1)) +
geom_line() + coord_cartesian(ylim=c(-10,10)) + facet_wrap(~group, ncol = 4,
scales = "free") + labs(title ='10-K Model Performance: Actual vs Model', sub
title = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year
Filed', caption = paste0('R-squared = ', market_agg_rsquared))

rm(market_agg, market_agg_rsquared, market_agg_actual, market_agg_predicted)
```



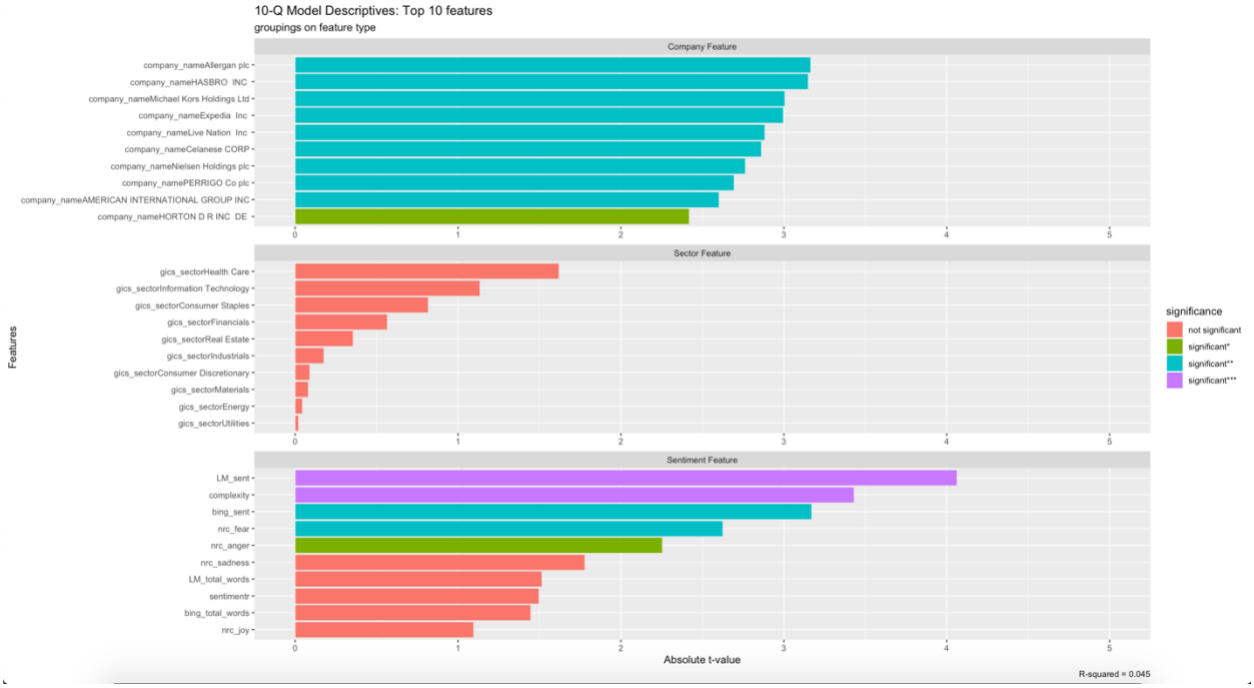
It is more fitted when brought to the S&P 500 market level with an explainatory power of 0.54.

Modelling: 10-Q Building Exhaustive Model to predict Stock Price Change

Same procedure in 10-K is used for 10-Q

```
load(file = '/Volumes/Buku Gibran/edgar/temp/sentiment_data.rda')
# ----- Stock Price Modelling
model_data_10q <- sentiment_data %>%
  filter(form_type == "10-Q") %>%
  select(-c(form_type, cik, return_adjusted_price))

accession_number <- model_data_10q$accession_number
year_filed <- model_data_10q$year_filed
model_data_10q <- model_data_10q %>% select(-c(year_filed, accession_number))
# year_filed is removed so it is not considered by the model

model_10q <- lm(price_adjusted_ratio ~., data = model_data_10q, na.action=na.
exclude)

save(model_10q, file = '/Volumes/Buku Gibran/edgar/temp/model_10q.rda')

load(file = '/Volumes/Buku Gibran/edgar/temp/model_10q.rda')
model_data_10q$predicted_ratio <- stats::predict(model_10q, newdata = model_d
ata_10q %>% select(-c(price_adjusted_ratio)))
model_data_10q$year_filed <- year_filed # add back year_filed as to help grou
ping
model_data_10q$accession_number <- accession_number
```

```r
save(model_data_10q, file = '/Volumes/Buku Gibran/edgar/temp/model_data_10q.rda')
load(file = '/Volumes/Buku Gibran/edgar/temp/model_data_10q.rda')


summary(model_10q)

local_df <- head(as.data.frame(summary(model_10q)$coefficients) %>%
                   tibble::rownames_to_column() %>%
                   mutate(absolute_t_value = abs(`t value`)) %>%
                   arrange(desc(absolute_t_value)) , 1000) %>%
  mutate(rowname=factor(rowname, levels=rowname)) %>%
  mutate(significance = case_when(`Pr(>|t|)` <= 0.001 ~ 'significant***', `Pr(>|t|)` <= 0.01 ~ 'significant**', `Pr(>|t|)` <= 0.05 ~ 'significant*', TRUE ~ 'not significant')) %>%
  mutate(r_squared = paste('Multiple R-squared:',as.character(round(summary(model_10q)$r.squared,3)))) %>%
  mutate(category = case_when(grepl("company_name", rowname, fixed = TRUE) ~ 'Company Feature', grepl("gics_sector", rowname, fixed = TRUE) ~ 'Sector Feature', TRUE ~ 'Sentiment Feature'))


feature_category <- c('Sentiment Feature', 'Sector Feature', 'Company Feature')

model_df <- data.frame()
for(v in 1:length(feature_category)) {
  top_n <- local_df %>% filter(category == feature_category[v]) %>% arrange(desc(absolute_t_value))
  top_n <- top_n[1:10,]
  model_df <- bind_rows(model_df, top_n)
}


ggplot(model_df, aes(x = reorder(rowname, absolute_t_value), y = absolute_t_value, fill=significance)) + geom_bar(stat = "identity") +
  labs(title ='10-Q Model Descriptives: Top 10 features', subtitle = 'groupings on feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squared = ', as.character(round(summary(model_10q)$r.squared,3)) )) +
  ylim(0,5) +  coord_flip() +
  facet_wrap(~category, nrow = 4, scales = "free")


rm(model_df, local_df, top_n, sentiment_data)
```

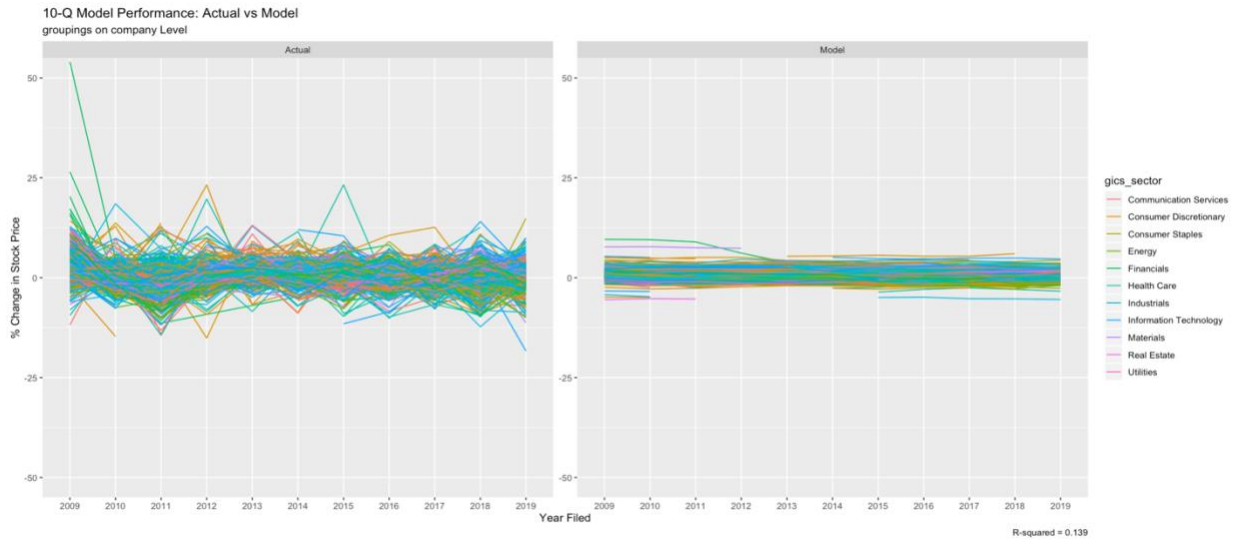


In contrast to 10-K's model, LM sentiment and complexity have bigger influence in 10-Q's model.

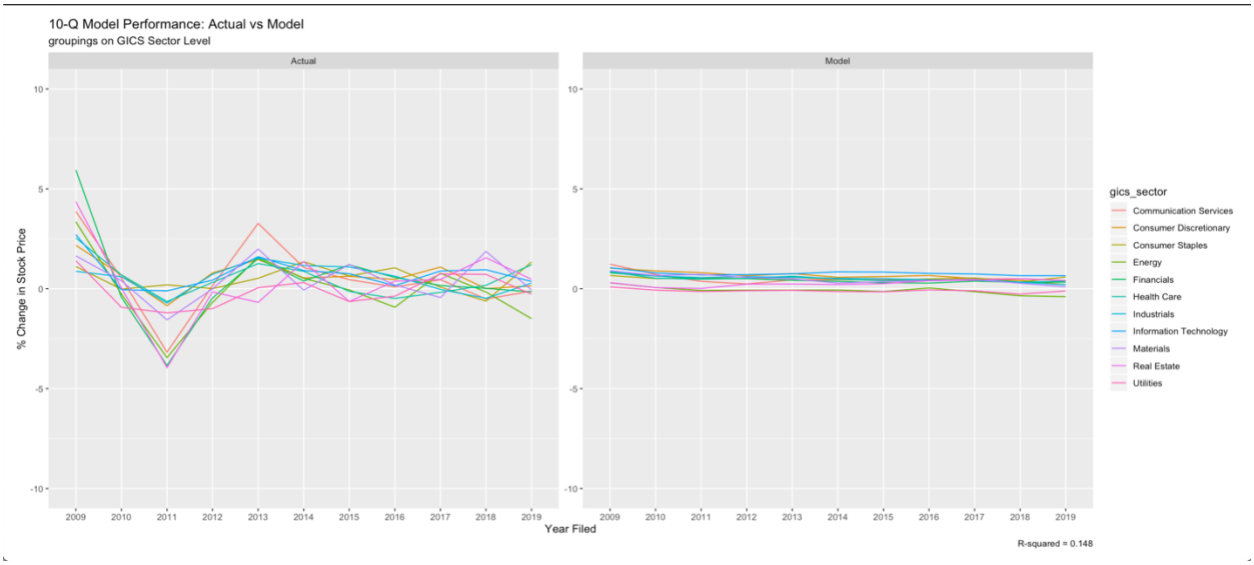
Model Evaluation - 10-Q

```
# ----- Company Level
company_agg <-  model_data_10q %>%
     group_by(year_filed, company_name, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

company_agg_rsquared <- round(rsq(company_agg$actual_ratio, company_agg$predi
cted_ratio), 3) # calculate R-squared between actual and predicted

company_agg_actual <- company_agg %>% select(-predicted_ratio) %>% mutate(pri
ce_adjusted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
company_agg_predicted <- company_agg %>% select(-actual_ratio) %>% mutate(pri
ce_adjusted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_r
atio)

company_agg <- bind_rows(company_agg_actual, company_agg_predicted)

# ----- Plot Actual vs Model
ggplot(company_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gic
s_sector)) + geom_line(aes(group = company_name)) + coord_cartesian(ylim=c(-5
0,50)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Mo
del Performance: Actual vs Model', subtitle = 'groupings on company Level',y
= '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared =
```

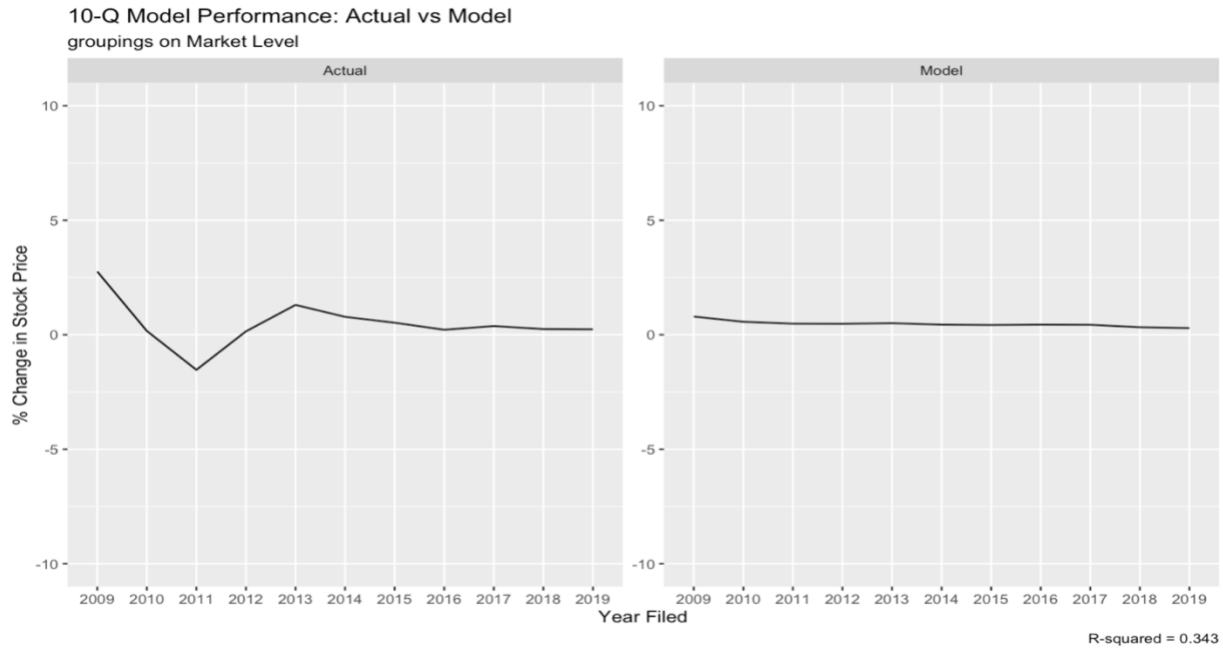
```
', company_agg_rsquared))

rm(company_agg, company_agg_rsquared, company_agg_actual, company_agg_predicted)
```




```
# ----- GICS Industry Level
gics_agg <-  model_data_10q %>%
     group_by(year_filed, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

gics_agg_rsquared <- round(rsq(gics_agg$actual_ratio, gics_agg$predicted_ratio), 3) # calculate R-squared between actual and predicted

gics_agg_actual <- gics_agg %>% select(-predicted_ratio) %>% mutate(price_adjusted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
gics_agg_predicted <- gics_agg %>% select(-actual_ratio) %>% mutate(price_adjusted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_ratio)

gics_agg <- bind_rows(gics_agg_actual, gics_agg_predicted)

# ----- Plot Actual vs Model
ggplot(gics_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gics_sector)) + geom_line(aes(group = gics_sector)) + coord_cartesian(ylim=c(-10,10)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Model Performance: Actual vs Model', subtitle = 'groupings on GICS Sector Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', gics_agg_rsquared))

rm(gics_agg, gics_agg_rsquared, gics_agg_actual, gics_agg_predicted)
```




```r
# ----- Market Level
market_agg <-  model_data_10q %>%
     group_by(year_filed) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              predicted_ratio = mean(predicted_ratio))

market_agg_rsquared <- round(rsq(market_agg$actual_ratio, market_agg$predicted_ratio), 3) # calculate R-squared between actual and predicted

market_agg_actual <- market_agg %>% select(-predicted_ratio) %>% mutate(price_adjusted_ratio = actual_ratio, group = 'Actual') %>% select(-actual_ratio)
market_agg_predicted <- market_agg %>% select(-actual_ratio) %>% mutate(price_adjusted_ratio = predicted_ratio, group = 'Model') %>% select(-predicted_ratio)

market_agg <- bind_rows(market_agg_actual, market_agg_predicted)

# ----- Plot Actual vs Model
ggplot(market_agg, aes(x = year_filed, y = price_adjusted_ratio, group =1)) + geom_line() + coord_cartesian(ylim=c(-10,10)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Model Performance: Actual vs Model', subtitle = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', market_agg_rsquared))

rm(market_agg, market_agg_rsquared, market_agg_actual, market_agg_predicted)
```

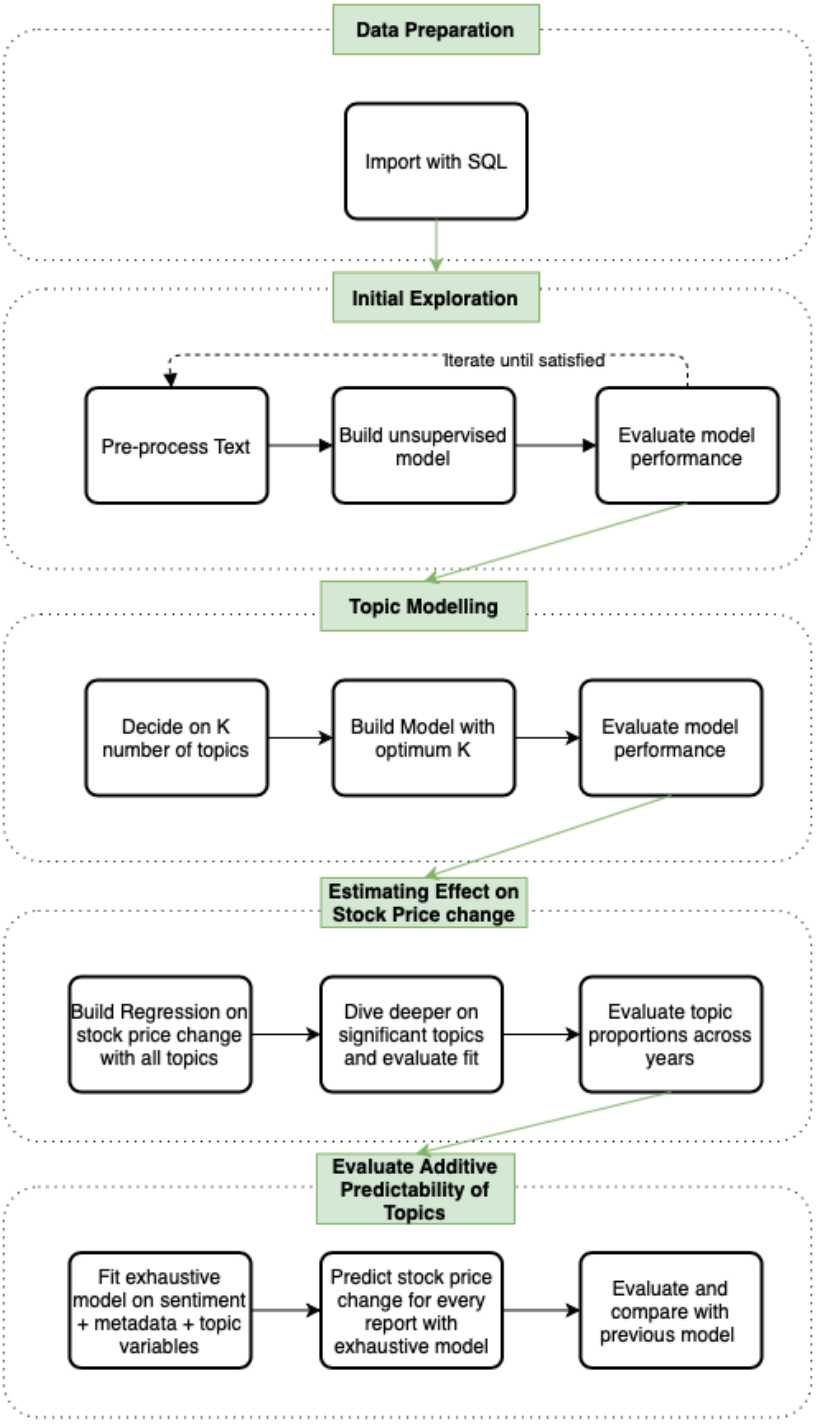


Conclusions

This section has proved that different sentiment features have different effect towards the % change in stock price depending on which sector it is. Although sentiment features are generally not as impactful compared to the meta-data of the reports, when used together it could reach to as much as an R-squared of 0.54 on the market level for 10-K reports. Spoilers alert! The R-squared could still be improved once variables from topic models in part C are added to the exhaustive model. So, let's jump to topic modelling.

Part C — Topic Modelling

Process Workflow



Data Preparation

Due to resource limitation, topic modelling was only conducted on 10-K reports. Thus, making this chapter focuses more on depth rather than volume of analysis.

```
conn <- dbConnect(RSQLite::SQLite(), "/Volumes/Buku Gibran/edgar/edgar.db")

documents_dataset <- dbGetQuery(conn, 'SELECT master_index.cik, company_name, gics_sector, form_type, date_filed, year_filed, accession_number, cleaned_noun, price_adjusted_ratio FROM master_index INNER JOIN sp500 ON sp500.cik = master_index.cik')  %>% mutate(date_filed = as.Date(date_filed, origin="1970-01-01")) %>% mutate(cik = as.factor(cik), company_name = as.factor(company_name), gics_sector = as.factor(gics_sector), form_type = as.factor(form_type), year_filed = as.factor(year_filed), accession_number = as.factor(accession_number))


documents_10q <- documents_dataset %>% filter(form_type == '10-Q')
save(documents_10q, file = '/Volumes/Buku Gibran/edgar/temp/documents_10q.rda')
load('/Volumes/Buku Gibran/edgar/temp/documents_10q.rda')

rm(documents_10q)

documents_10k <- documents_dataset %>% filter(form_type == '10-K')
save(documents_10k, file = '/Volumes/Buku Gibran/edgar/temp/documents_10k.rda')


documents_10k <- documents_10k %>% na.omit()
load('/Volumes/Buku Gibran/edgar/temp/documents_10k.rda')
rm(documents_dataset)
```

Initial Exploration (Unsupervised Approach)

The aim of topic modelling is to discover themes or topics assumed to have in a corpus of documents. In specfic, this section will approach topic modelling using the Structural Topic Modelling that can allow researchers to estimate relationships between topics and its document's meta-data (Robert, Stewart, & Tingley, 2016). The initial exploration will be conducted in iterations until meaningful topics emerges.

There are two main goals of initial exploration; First, to identify appropriate K (kappa) number of topics. Second, to check whether another iteration of data cleaning is required to achieve a reasonable set of topics.

Pre-process Text and Building Unsupervised Model

By setting K = 0 and init.type = 'Spectral' the stm function will automatically select the number of topics, although there is no guarantee of achieving optimum, but it is claimed to be a good start as it has a good computational advantage since it only need to run once (Robert, Stewart, & Tingley, 2016).

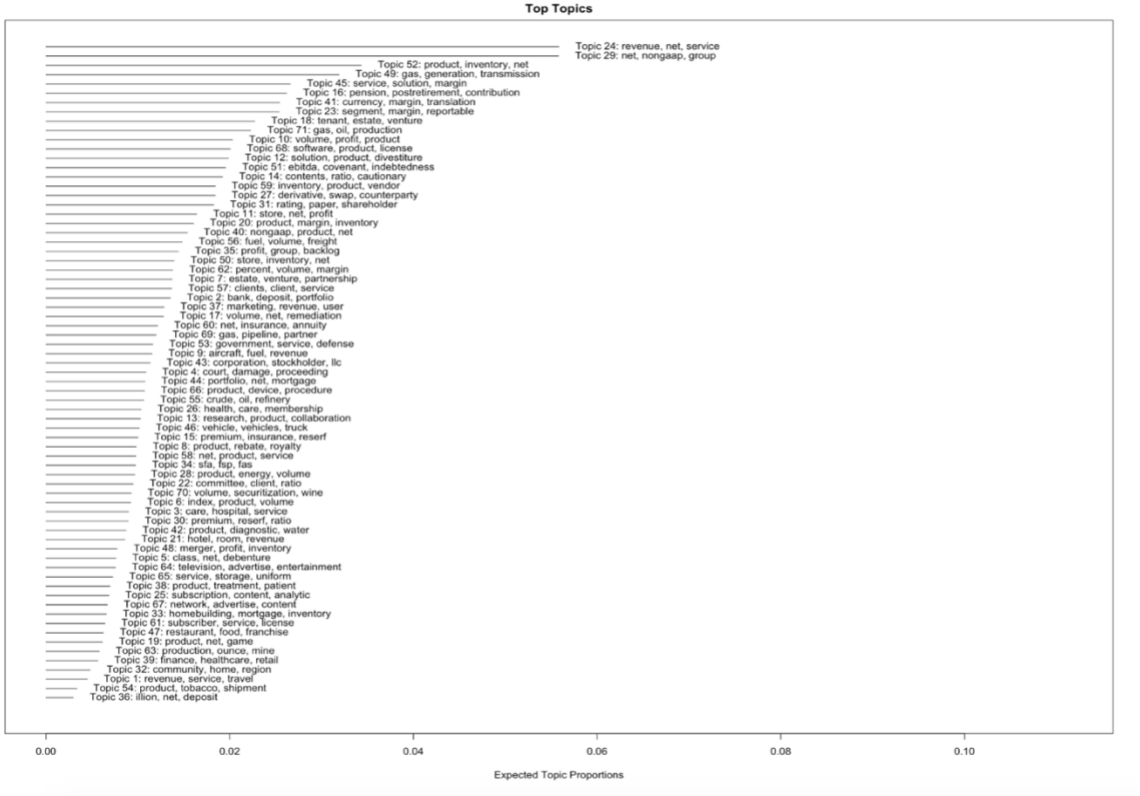
```
set.seed(2107)

# ----- Corpus Preparation
processed <- textProcessor(sample_10k$cleaned_noun,
                           metadata = sample_10k,
                           customstopwords = c("net","product","service","mar
gin","volume","revenue","inventory"),
                           stem = F)

threshold <- round(1/100 * length(processed$documents),0)

out_10k <- prepDocuments(processed$documents,
                     processed$vocab,
                     processed$meta,
                     lower.thresh = threshold)

# ----- STM model fitting
stm_unsupervised_10k <- stm(documents = out_10k$documents,
                      vocab = out_10k$vocab,
                      K = 0,
                      prevalence = NULL,
                      max.em.its = 150,
                      data = out_10k$meta,
                      reportevery = 5,
                      sigma.prior = 0.7,
                      init.type = "Spectral")

save(stm_unsupervised_10k, file = '/Volumes/Buku Gibran/edgar/temp/stm_unsupe
rvised_10k.rda')
```

Evaluate Model Performance

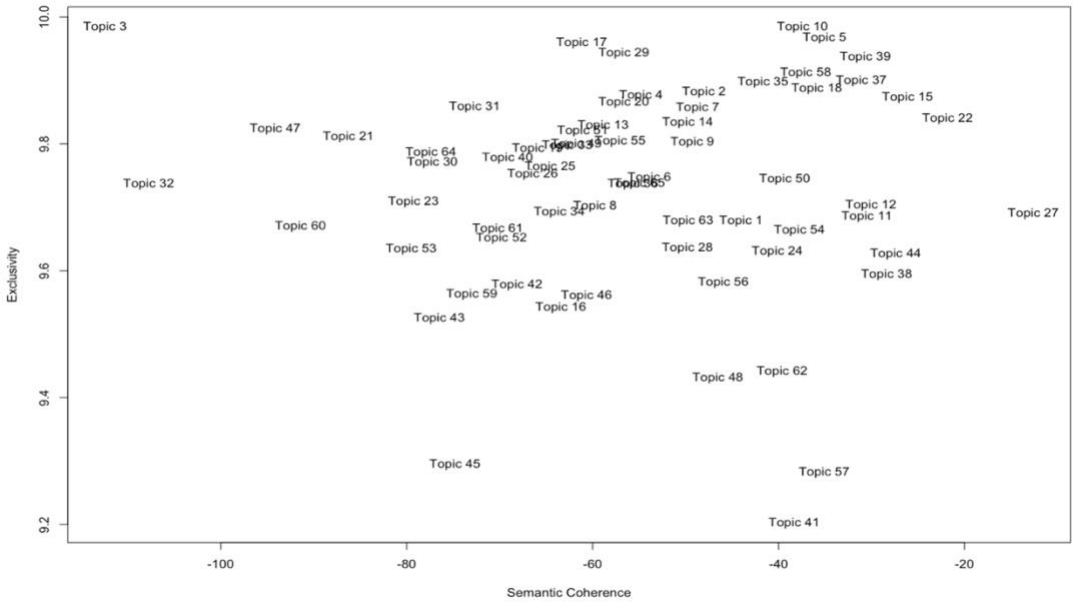
By reviewing the summary of the unsupervised stm model, we can review the proposed topics including its top 7 words arranged by FREX (overall frequency and how exlcusive the words to that topic), lift weights (higher weight when less frequent in other topics), and probablity of the word belong to the topic (Robert, Stewart, & Tingley, 2016). Although it sounds less complex than FREX and lift, I would argue that the word-topic probability is what we are looking for to identify words that 'cannot make their mind to which topic they belong', or simply put, stop words.

```r
# ----- Review performance
topic_summary_unsupervised <- summary(stm_unsupervised_10k)
plot(stm_unsupervised_10k) # plot the topic model
topicQuality(stm_unsupervised_10k,documents = out_10k$documents) # review top
ic semantic-coherence

# ----- Review word frequency to identify potential stopwords

unsupervised_k <- length(topic_summary_unsupervised$topicnums)
top_topic_words <- c()
for (i in 1:unsupervised_k){
  top_topic_words <- c(top_topic_words, topic_summary_unsupervised$prob[i,])
}

data.frame(word = top_topic_words) %>%
  group_by(word) %>%
  summarise(count =n()) %>%
  arrange(desc(count)) %>%
  top_n(50) %>%
  mutate(word = factor(word, word)) %>%
  ggplot(aes(x = reorder(word, count), y = count)) + geom_bar(stat="identity"
) + coord_flip() + labs(title = 'Occurance of Highest Probable Words Across T
opics',subtitle = paste('a result of unsupervised stm with', unsupervised_k,
'number of topics'), x ='Word') + scale_y_continuous(breaks=c(2,4,6,8,10))



# ----- Review FREX words for every topic
topic_proportions <- colMeans(stm_unsupervised_10k$theta)

unsupervised_frex <- data.frame()
for(i in 1:length(topic_summary_unsupervised$topicnums)){

   row_here <- tibble(topicnum= topic_summary_unsupervised$topicnums[i],
                  #    topic_label = topic_labels[i],
                      proportion = 100*round(topic_proportions[i],4),
                     frex_words = paste(topic_summary_unsupervised$frex[i,1:7
],
                                        collapse = ","))
   unsupervised_frex <- rbind(row_here,unsupervised_frex)
}


unsupervised_frex %>%
  arrange(desc(proportion))
```

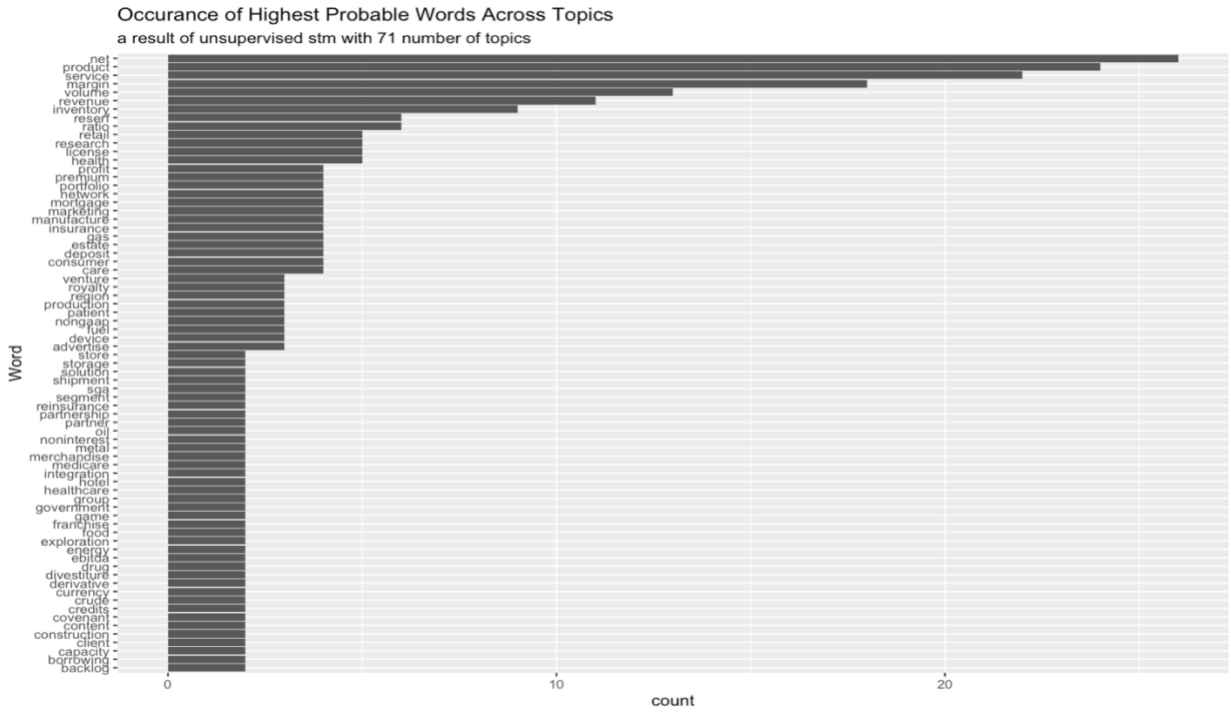
First Iteration – without additional stop-words



The model suggests 71 number of topics. However, there seems to be some overlaps.

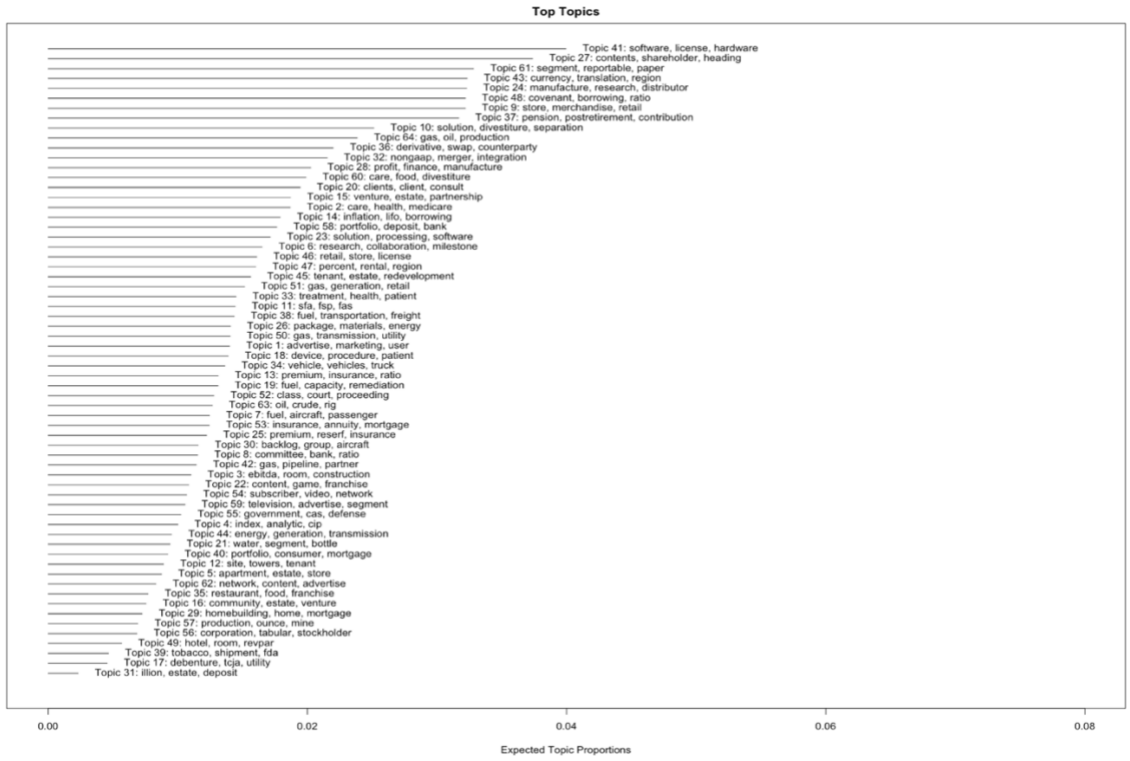


The semantic coherence and exclusivity of the model quality is fairly good.

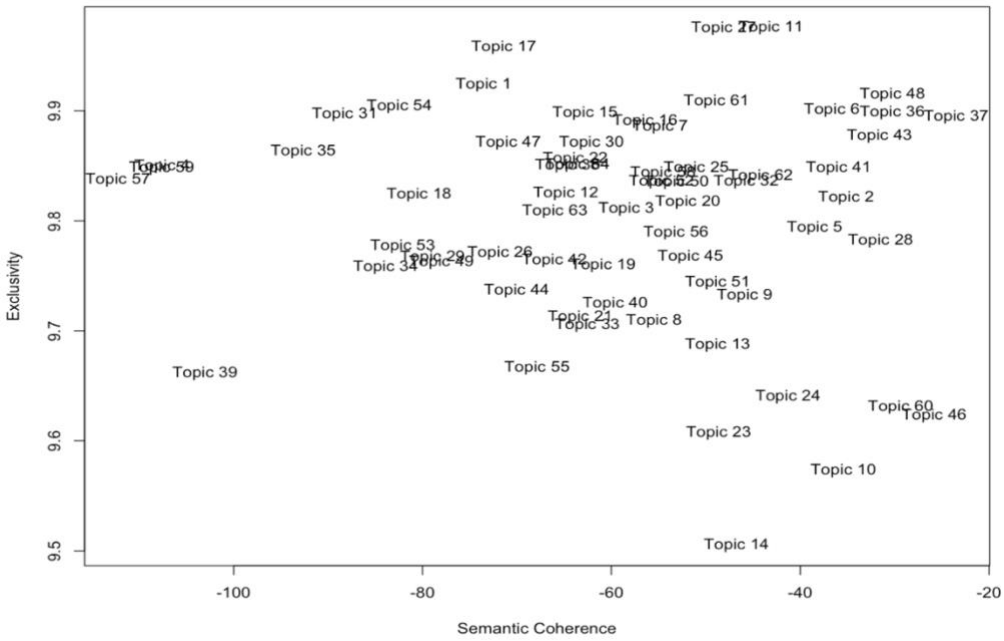


By plotting the frequency of the most probable words occuring in multiple topics, we can identify potential stop words. Author decided to consider the top 7 words as stop words, given its extreme 'deep-dive' distribution (p.s. zipf's law).

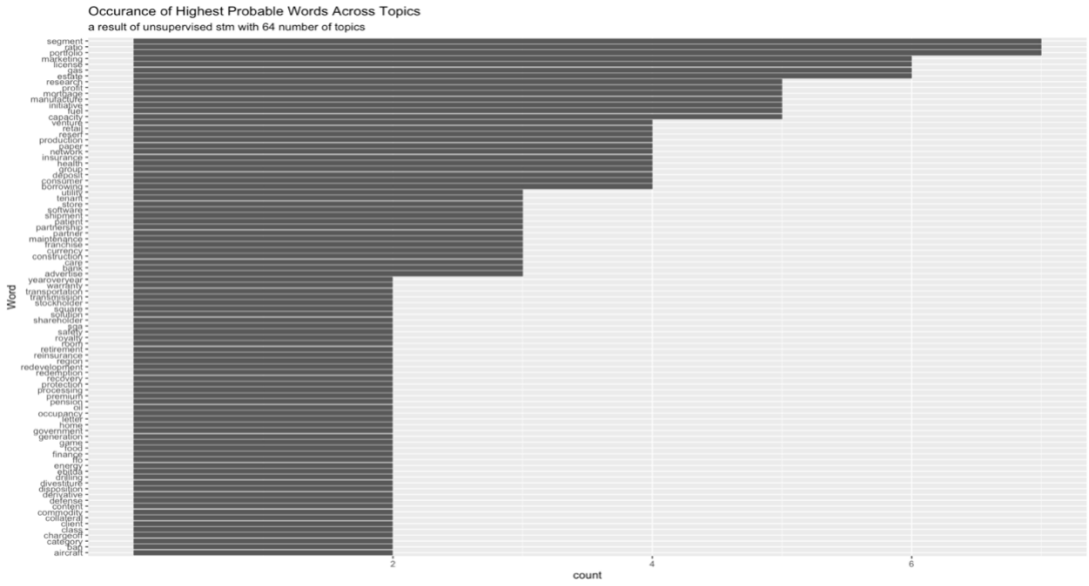
Second Iteration – after addition of stop-words



In the second iteration, the unsupervised model suggests lower topic number of 64. With the highest expected topic proportion around 0.04 on topic 41 which seems to be related to the information technology industry.



The exclusivity of the topics has improved to the point where the minimum exclusivity of a topic is over 9.5, in contrast to the previous iteration of 9.2. The topics also seem to converge towards the right handside proving superior quality in semantic coherence.



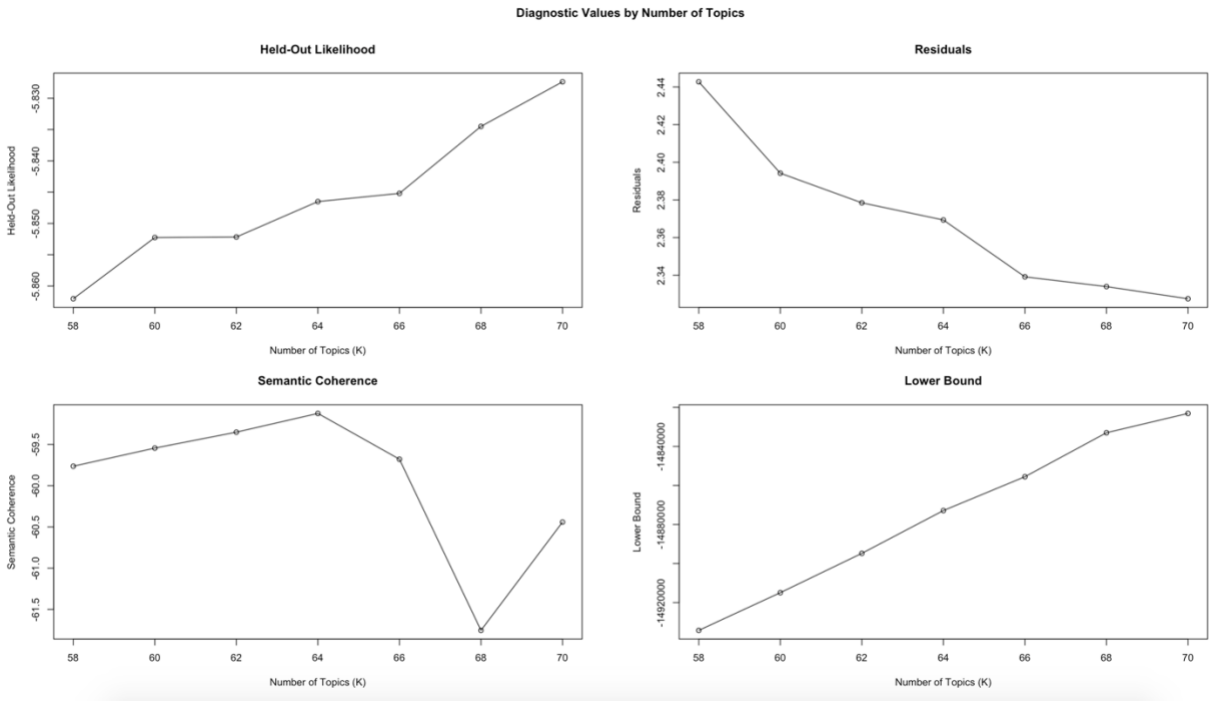
The distribution has now seem to be more controlled. Therefore, it is assumed that the remaining words are non-stop words, thus the initial exploration stops with a best K of 64 and 7 additional stop words.

Modelling (Supervised Approach)

Deciding on K number of topics

The decision to pick the number of topics, Kappa (K), is guided with a heuristics by defining a neighbourhood around the best K known from the initial exploration (unsupervised approach). The neighbourhood is defined as 3 K neighbours to the left and 3 K to the right with an interval of 2 between K.

```
# ----- SearchK
expected_K <- c(unsupervised_k - 6 , unsupervised_k -4 ,unsupervised_k - 2, u
nsupervised_k, unsupervised_k + 2 ,unsupervised_k+4, unsupervised_k+6) #unsup
ervised_k is 64
sk_result <- searchK(out_10k$documents,out_10k$vocab, expected_K)
save(sk_result, file = '/Volumes/Buku Gibran/edgar/temp/sk_result.rda')

plot(sk_result)
```



Since the held-out likelihood is highest and the residuals is lowest at K = 70, it is an easy decision to set Kappa as 70 for the final topic modelling using the supervised approach.

Build Supervised Topic Model with optimum K

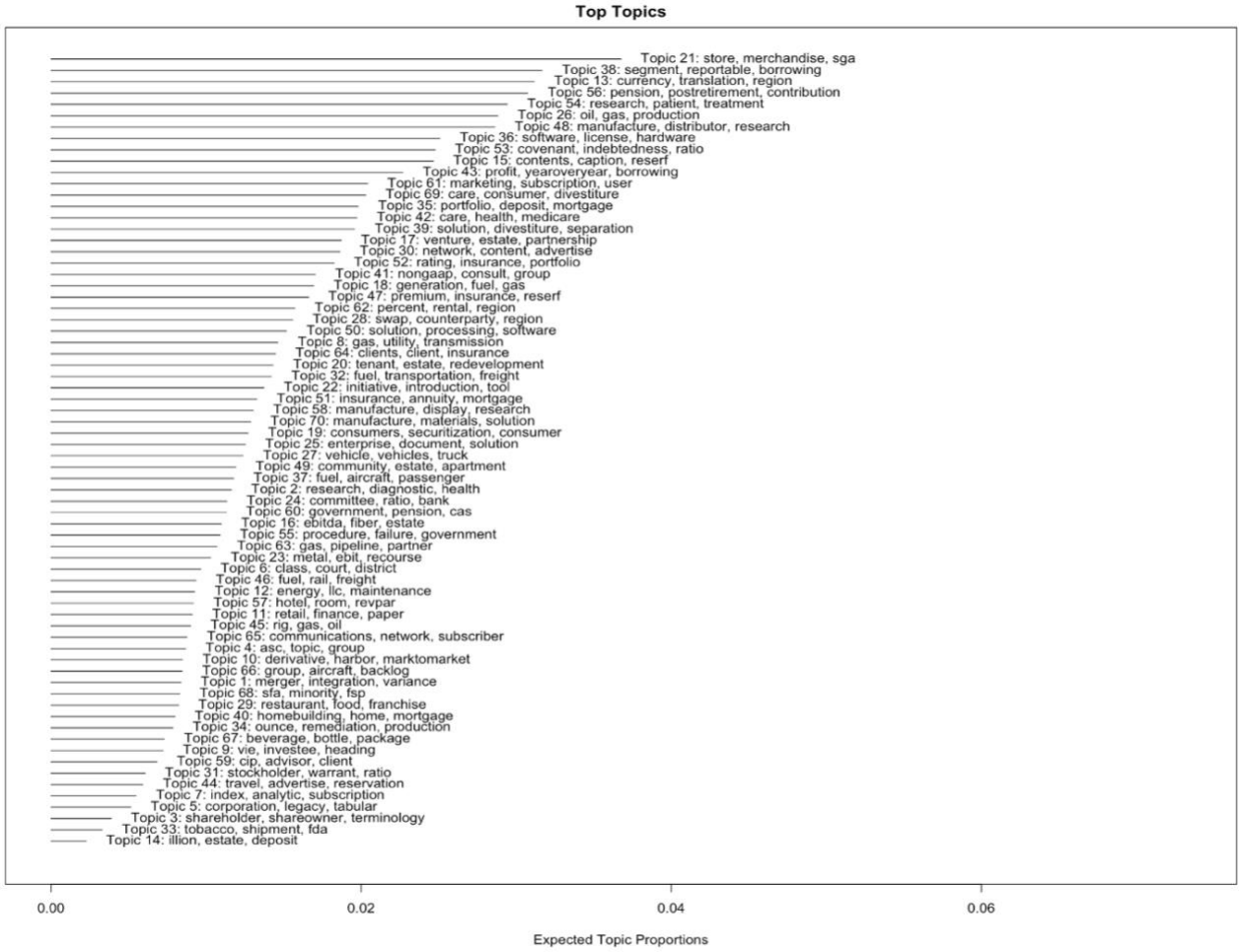
```
set.seed(2107)

# ----- Corpus Preparation
processed <- textProcessor(documents_10k$cleaned_noun,
```

```
                            metadata = documents_10k,
                            customstopwords = c("net","product","service","mar
gin","volume","revenue","inventory"),
                            stem = FALSE)

threshold <- round(1/100 * length(processed$documents),0)

out_10k <- prepDocuments(processed$documents,
                     processed$vocab,
                     processed$meta,
                     lower.thresh = threshold)
save(out_10k, file = '/Volumes/Buku Gibran/edgar/temp/out_10k.rda')

# ----- STM model fitting
stm_supervised_10k <- stm(documents = out_10k$documents,
                      vocab = out_10k$vocab,
                      K = 70,
                      prevalence = ~ factor(gics_sector) + s(year_filed),
                      max.em.its = 150,
                      data = out_10k$meta,
                      reportevery = 5,
                      sigma.prior = 0.7,
                      init.type = "Spectral")

save(stm_supervised_10k, file = '/Volumes/Buku Gibran/edgar/temp/stm_supervis
ed_10k.rda')

topic_summary_supervised <- summary(stm_supervised_10k)
```

Evaluate Supervised Model Performance

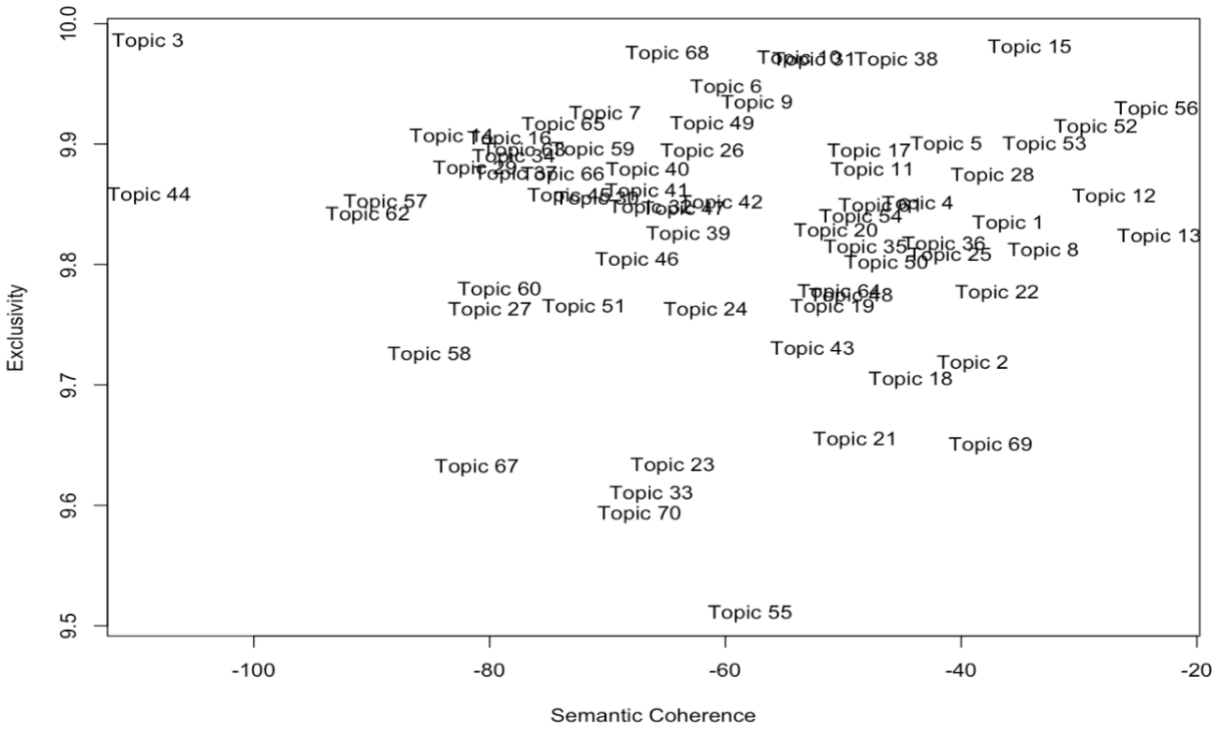
```
# ----- Review performance
plot(stm_supervised_10k) # plot the topic model
topicQuality(stm_supervised_10k,documents = out_10k$documents) # review topic
semantic-coherence


# ----- Review word frequency to identify potential stopwords
top_topic_words <- c()
supervised_k <- length(topic_summary_supervised$topicnums)

for (i in 1:supervised_k){
  top_topic_words <- c(top_topic_words, topic_summary$prob[i,])
}

data.frame(word = top_topic_words) %>%
  group_by(word) %>%
  summarise(count =n()) %>%
  arrange(desc(count)) %>%
```

```
  top_n(50) %>%
  mutate(word = factor(word, word)) %>%
  ggplot(aes(x = reorder(word, count), y = count)) + geom_bar(stat="identity"
) + coord_flip() + labs(title = 'Occurance of Highest Probable Words Across T
opics',subtitle = paste('a result of supervised stm with', supervised_k, 'num
ber of topics'), x ='Word')


# ----- Review FREX words for every topic
topic_proportions <- colMeans(stm_supervised_10k$theta)

supervised_frex <- data.frame()
for(i in 1:length(topic_summary_supervised$topicnums)){

   row_here <- tibble(topicnum= topic_summary_supervised$topicnums[i],
                  #    topic_label = topic_labels[i],
                      proportion = 100*round(topic_proportions[i],4),
                     frex_words = paste(topic_summary_supervised$frex[i,1:7],
                                        collapse = ","))
   supervised_frex <- rbind(row_here,supervised_frex)
}
rm(row_here)

supervised_frex %>%
  arrange(desc(proportion)) %>%
  filter(topicnum == 68)
```



In comparison to the previous two unsupervised set of topics, the topic set suggested by the supervised STM model seem to have more obviously related words. Some instances includes:

- **Topic 45** — rig, gas, oil [Energy]
- **Topic 54** — research, patient, treatment [Health care]
- **Topic 36** — software, license, hardware [IT]
- **Topic 56** — retirement, postretirement, contribution [Retirement]
- **Topic 61** — marketing, subscription, user [Subscription businesses]
- **Topic 37** — fuel, airline, passengar [Airline]
- **Topic 57** — hotel, room, revpar (revenue per available room) [Hotels]
- **Topic 1** — merger, integration, variance [Mergers & Acquisitions]
- **Topic 29** — restaurant, food, franchise [Food & Beverages]
- **Topic 44** — travel, advertise, reservation [Hospitality & Travel]



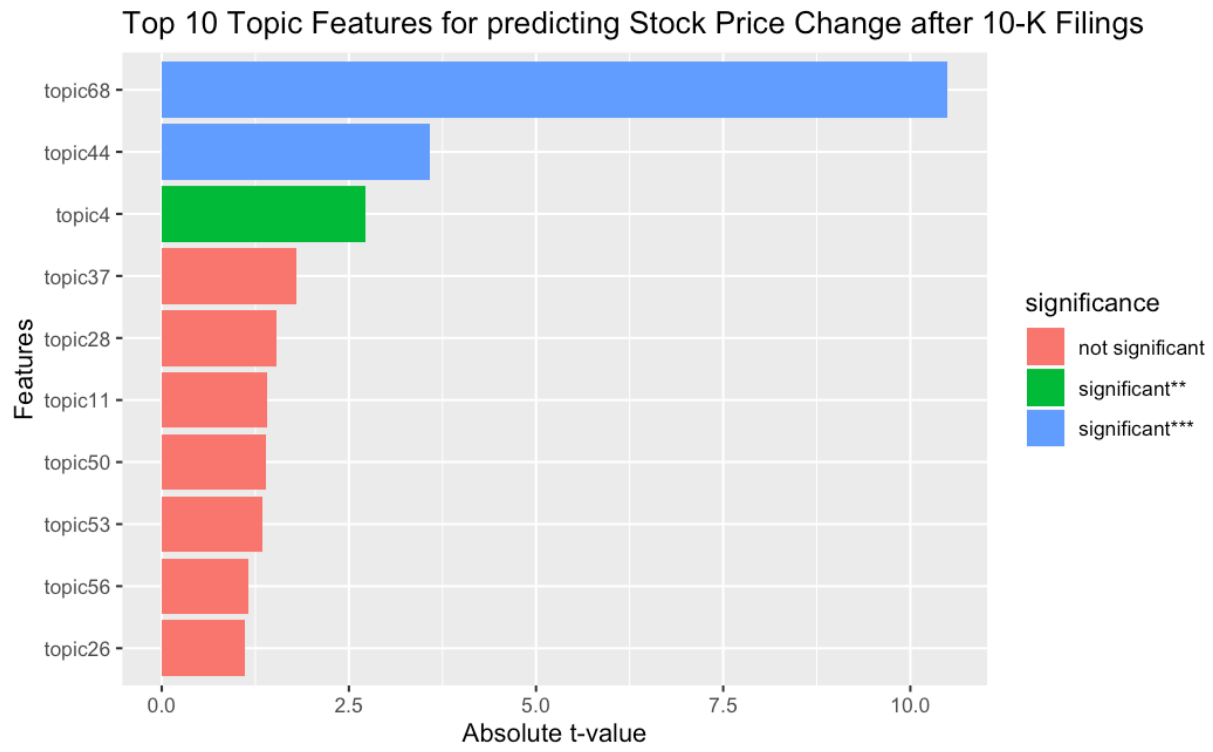
Additionally, there are more topics that roams around to almost 10 in exclusivity score, resulting to more distinctive topics. While on the other hand more topics are converging to the right upper hand-side suggesting that the words within a topic are more correlated to each other. Thus, making the supervised model more superior than the previous two topic sets.

Estimating Effect: How Topic affect Stock Price Change?

Similar to what we did in sentiment analysis, a regression model is fit on all topic variables to predict stock price % change.

```
# ----- All topics effect
convergence <- as.data.frame(stm_supervised_10k$theta)
colnames(convergence) <- paste0("topic",1:70)

regression_data <- cbind(out_10k$meta,convergence) %>% na.omit() %>% select(-
c(cleaned_noun, accession_number, date_filed, cik, year_filed, company_name,
gics_sector, form_type))

topic_regression <- lm(price_adjusted_ratio ~ . ,data = regression_data)
```

```r
topic_regression_summary <- head(as.data.frame(summary(topic_regression)$coef
ficients) %>%
    tibble::rownames_to_column() %>%
    mutate(absolute_t_value = abs(`t value`)) %>%
    arrange(desc(absolute_t_value)) , 10) %>%
    mutate(rowname=factor(rowname, levels=rowname)) %>%
    mutate(significance = case_when(`Pr(>|t|)` <= 0.001 ~ 'significant***', `
Pr(>|t|)` <= 0.01 ~ 'significant**', `Pr(>|t|)` <= 0.05 ~ 'significant*', TRU
E ~ 'not significant')) %>%
    mutate(r_squared = paste('Multiple R-squared:',as.character(round(summary
(topic_regression)$r.squared,3))))

ggplot(topic_regression_summary, aes(x = reorder(rowname, absolute_t_value),
y = absolute_t_value, fill=significance)) + geom_bar(stat = "identity") +
  labs(title ='Top 10 Topic Features for predicting Stock Price Change after
10-K Filings', y = 'Absolute t-value', x = 'Features') + coord_flip()
```




```r
# ----- Topic effect singualar
library(stargazer)
topic68_regression <- lm(price_adjusted_ratio ~ topic68,data = regression_dat
```

```
a)
topic44_regression <- lm(price_adjusted_ratio ~ topic44,data = regression_dat
a)
topic4_regression <- lm(price_adjusted_ratio ~ topic4,data = regression_data)
stargazer::stargazer(topic68_regression,topic44_regression,topic4_regression,
type = "text")

rm(topic_regression, topic_regression_summary, topic68_regression, topic44_re
gression, topic4_regression, regression_data)
```

```
=============================================================
                                 Dependent variable:
                             ----------------------------
                                 price_adjusted_ratio
                                   (1)       (2)       (3)
-------------------------------------------------------------
topic68                         -0.268***
                                 (0.023)

topic44                                    0.058***
                                           (0.013)

topic4                                                0.120***
                                                      (0.034)

Constant                         0.006***  0.004***  0.003***
                                 (0.001)    (0.001)   (0.001)

-------------------------------------------------------------
Observations                      4,718      4,718     4,718
R2                                0.028      0.004     0.003
Adjusted R2                       0.027      0.004     0.002
Residual Std. Error (df = 4716)   0.050      0.050     0.050
F Statistic (df = 1; 4716)      134.350*** 20.232*** 12.741***
=============================================================
Note:                             *p<0.1; **p<0.05; ***p<0.01
```

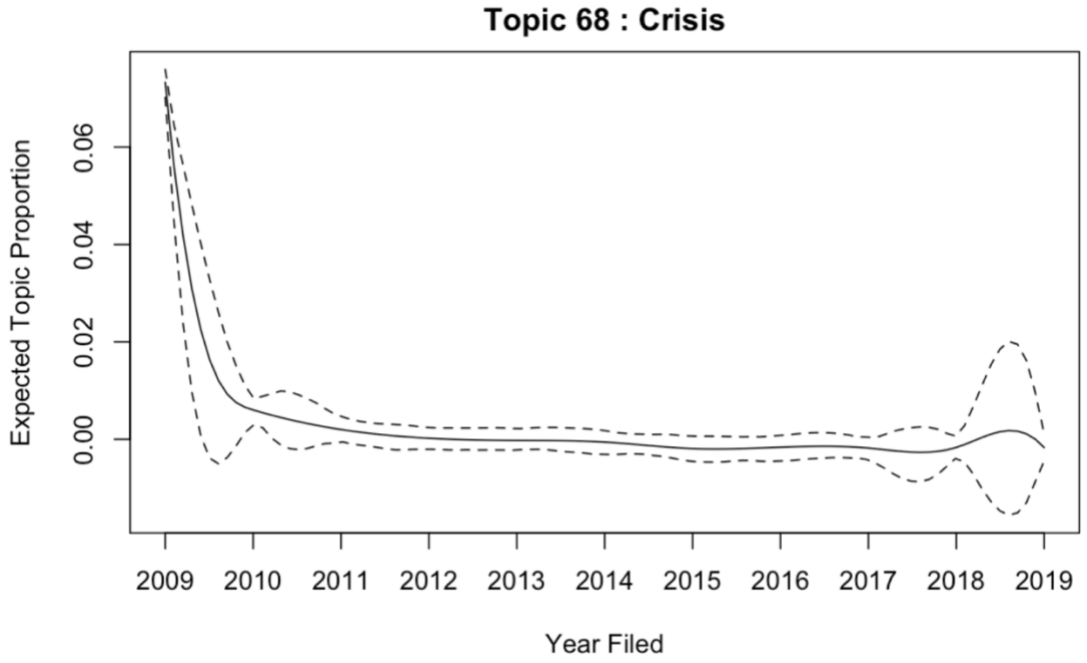
Topic 68, 44, and 4 significantly affect the stock price change. Topic 68 alone accounts to almost 3% variance of stock price change.

```
# ----- Plot most significant topics in cloud words
cloud(stm_supervised_10k, topic = 68, type = c("model"), max.words = 100)
cloud(stm_supervised_10k, topic = 44, type = c("model"), max.words = 100)
cloud(stm_supervised_10k, topic = 4, type = c("model"), max.words = 100)
```

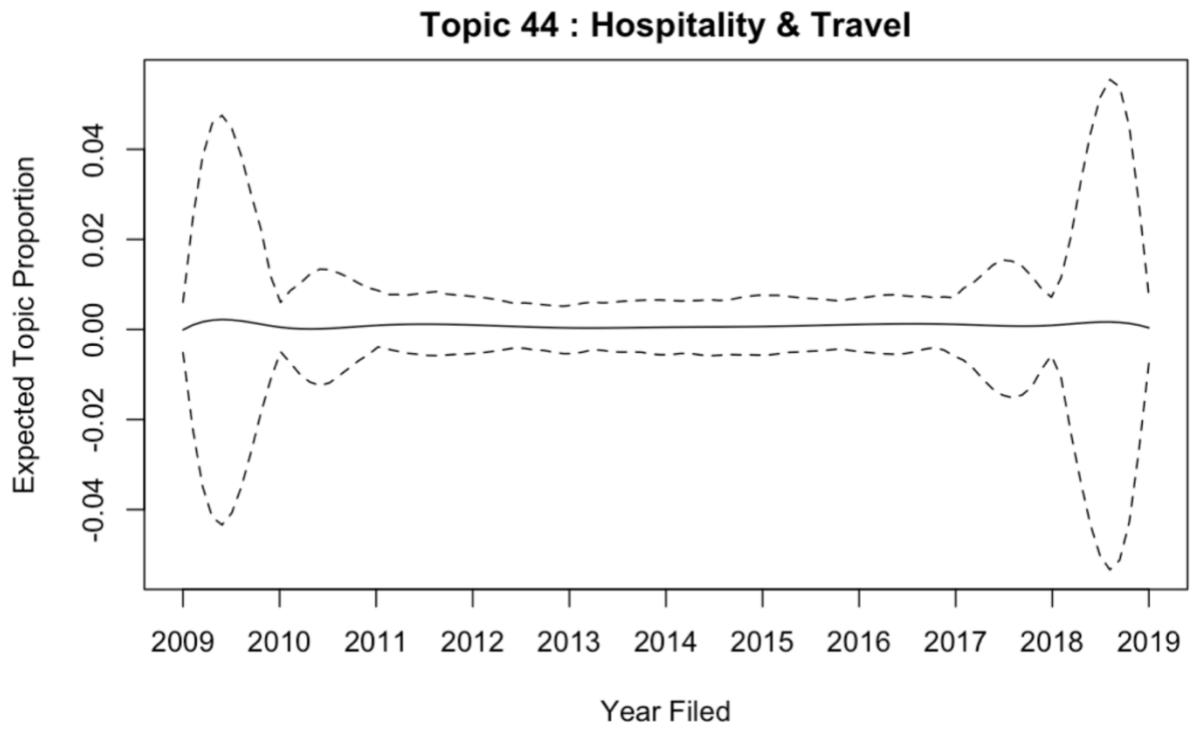
Estimate Effects: Topics Proportions over time

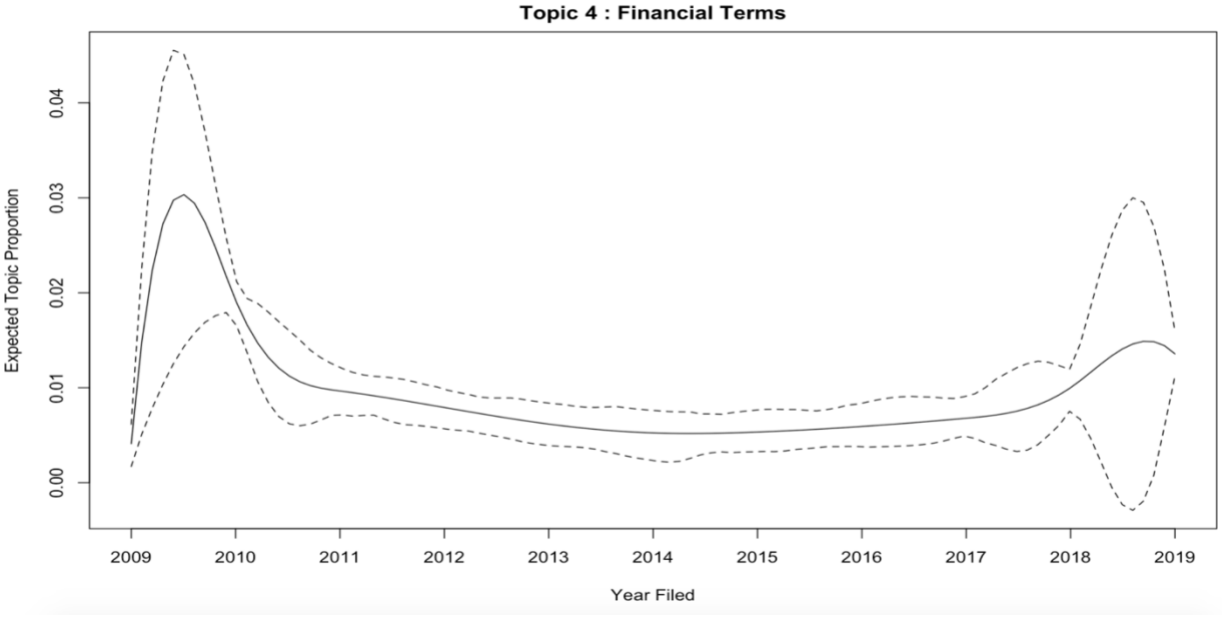
```
load('/Volumes/Buku Gibran/edgar/temp/stm_supervised_10k.rda')
load('/Volumes/Buku Gibran/edgar/temp/out_10k.rda')

out_10k$meta$year_filed <- as.numeric(out_10k$meta$year_filed)

effects_10k <- estimateEffect(~ factor(gics_sector) + s(year_filed),
                              stmobj = stm_supervised_10k,
                              metadata = out_10k$meta,
                              uncertainty = 'None')




convergence <- as.data.frame(stm_supervised_10k$theta)
colnames(convergence) <- paste0("topic",1:70)
```

```r
topics_of_interest <- c(68,44,4)
topic_labels <- c("Crisis","Hospitality & Travel", "Financial Terms")

for (t in 1:length(topics_of_interest)){

plot(effects_10k, covariate = "year_filed",
        topics = topics_of_interest[t],
        model = stm_supervised_10k, method = "continuous",
        xaxt='n',
        xlab="Year Filed",
        main = paste('Topic', topics_of_interest[t],':',topic_labels[t]),
        printlegend = FALSE,
        linecol = "black",
        labeltype = "none")

   axis(1,at=seq(from=1,
                 to= length(unique(out_10k$meta$year_filed)),
                 by=1),  labels= c('2009','2010','2011','2012','2013','2014',
'2015','2016','2017','2018','2019'))

}

out_10k$meta$year_filed <- as.factor(out_10k$meta$year_filed)
```



As confirmed in the TF-IDF analysis and the EDA in part B, 2009 seem to be the year which is different compared to other years. Interestingly, topic 68, which talks about crisis only peaks in 2009 too. A coincidence?


Topic 44 : Hospitality & Travel
Expected Topic Proportion
-0.04
-0.02
0.00
0.02
0.04
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year Filed


Topic 4 : Financial Terms
Expected Topic Proportion
0.00
0.01
0.02
0.03
0.04
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Year Filed

Evaluate Additive Predictability of Topics

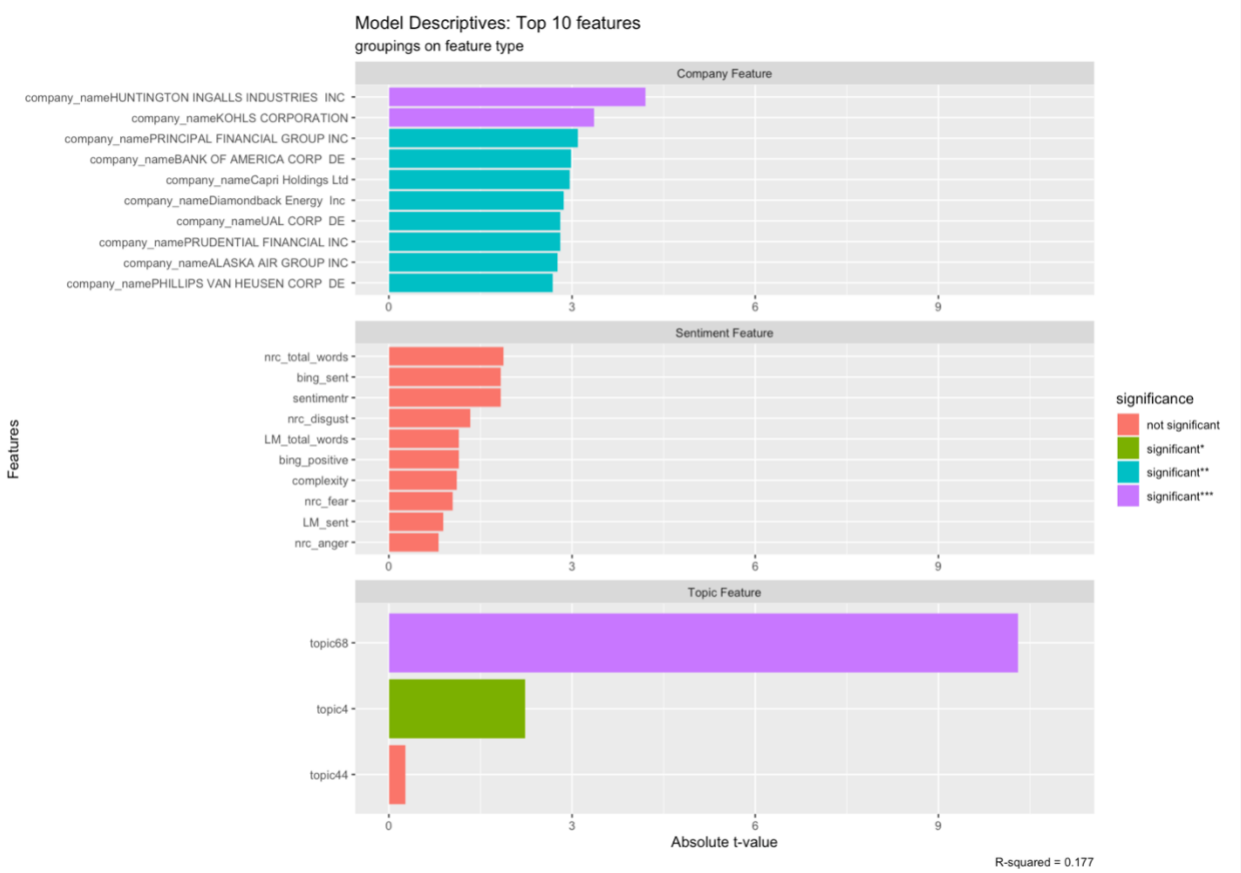
Model Fitting with Topic Features as addition

Now here comes the moment of truth, whether topic modelling could improve the exhaustive model's predictability discussed in part B. Only significant topics were added to the model.

```
load(file = '/Volumes/Buku Gibran/edgar/temp/model_data_10k.rda')
# ----- Adding significant topic variables to Exhaustive Model
regression_data_full <- cbind(out_10k$meta,convergence) %>% select(cik, compa
ny_name, accession_number, gics_sector, year_filed, topic68, topic44, topic4)
%>% na.omit() %>% left_join(model_data_10k %>% select(-c(year_filed, gics_sec
tor, company_name)), by= 'accession_number') %>% ungroup() %>% drop_na()

accession_number <- regression_data_full$accession_number
year_filed <- regression_data_full$year_filed
predicted_ratio <- regression_data_full$predicted_ratio
regression_data_full <- regression_data_full %>% select(-c(year_filed, access
ion_number, predicted_ratio, cik))

model_10k_with_topic <- lm(price_adjusted_ratio ~., data = regression_data_fu
ll, na.action=na.exclude)

summary(model_10k_with_topic)

regression_data_full$new_predicted_ratio <- stats::predict(model_10k_with_top
ic, newdata = regression_data_full %>% select(-c(price_adjusted_ratio)))
regression_data_full$year_filed <- year_filed # add back year_filed as to hel
p grouping
regression_data_full$accession_number <- accession_number
regression_data_full$prev_predicted_ratio <- predicted_ratio



save(model_10k_with_topic, file = '/Volumes/Buku Gibran/edgar/temp/model_10k_
with_topic.rda')
save(regression_data_full, file = '/Volumes/Buku Gibran/edgar/temp/regression
_data_full.rda')

load(file = '/Volumes/Buku Gibran/edgar/temp/model_10k_with_topic.rda')
load(file = '/Volumes/Buku Gibran/edgar/temp/regression_data_full.rda')
```

Model Evaluation

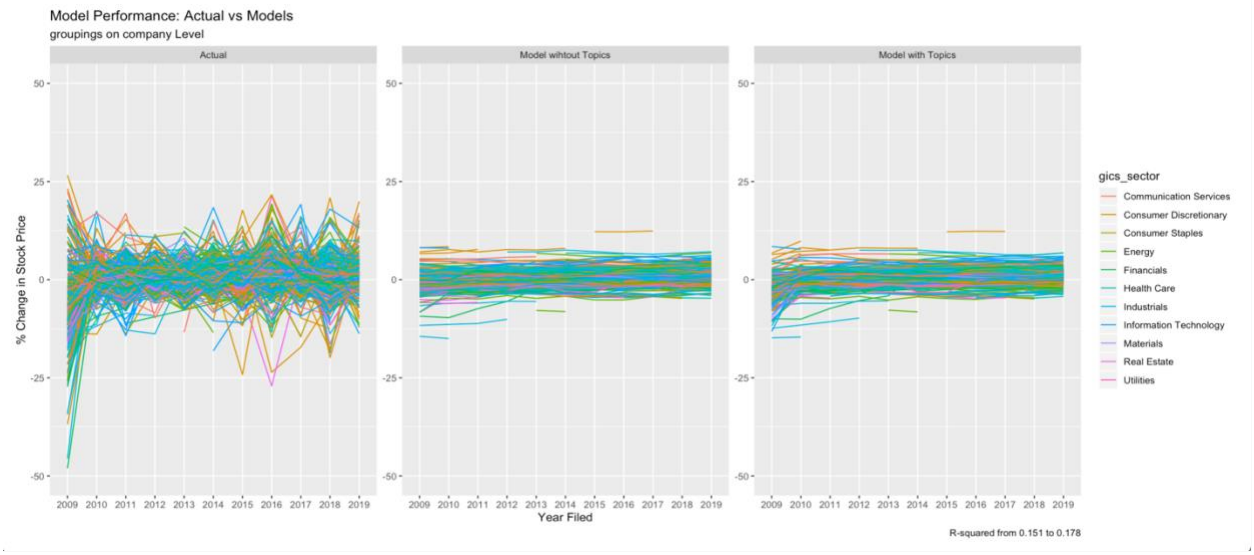
```
# ----- Evaluate Model
local_df <- head(as.data.frame(summary(model_10k_with_topic)$coefficients) %>
%
    tibble::rownames_to_column() %>%
    mutate(absolute_t_value = abs(`t value`)) %>%
    arrange(desc(absolute_t_value)) , 1000) %>%
    mutate(rowname=factor(rowname, levels=rowname)) %>%
    mutate(significance = case_when(`Pr(>|t|)` <= 0.001 ~ 'significant***', `
Pr(>|t|)` <= 0.01 ~ 'significant**', `Pr(>|t|)` <= 0.05 ~ 'significant*', TRU
E ~ 'not significant')) %>%
    mutate(r_squared = paste('Multiple R-squared:',as.character(round(summary
(model_10k_with_topic)$r.squared,3)))) %>%
    mutate(category = case_when(grepl("company_name", rowname, fixed = TRUE)
~ 'Company Feature', grepl("topic", rowname, fixed = TRUE) ~ 'Topic Feature',
TRUE ~ 'Sentiment Feature'))


feature_category <- c('Topic Feature','Sentiment Feature', 'Company Feature')

model_df <- data.frame()
for(v in 1:length(feature_category)) {
  top_n <- local_df %>% filter(category == feature_category[v]) %>% arrange(d
esc(absolute_t_value))
  top_n <- top_n[1:10,]
  model_df <- bind_rows(model_df, top_n) %>% drop_na()
}


ggplot(model_df, aes(x = reorder(rowname, absolute_t_value), y = absolute_t_v
alue, fill=significance)) + geom_bar(stat = "identity") +
  labs(title ='Model Descriptives: Top 10 features', subtitle = 'groupings on
feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squ
ared = ', as.character(round(summary(model_10k_with_topic)$r.squared,3)) )) +
coord_flip() + ylim(0,11) +
  facet_wrap(~category, nrow = 4, scales = "free")

rm(top_n, model_df, local_df)
```



In comparison to the 10-K model discussed previously where 4 out of 4 features which have significance of *** ($p < 0.01$) were all company features, now topic 68 dominates the model with over 9 absolute t-value.

```
rsq <- function (x, y) cor(x, y) ^ 2 # setup R-squared calculation

# ----- Company Level
company_agg <-  regression_data_full %>%
     group_by(year_filed, company_name, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              prev_predicted_ratio = mean(prev_predicted_ratio),
              new_predicted_ratio = mean(new_predicted_ratio))

company_agg_rsquared_prev <- round(rsq(company_agg$actual_ratio, company_agg$
prev_predicted_ratio), 3) # calculate R-squared between actual and predicted
company_agg_rsquared_new <- round(rsq(company_agg$actual_ratio, company_agg$n
ew_predicted_ratio), 3) # calculate R-squared between actual and predicted


company_agg_actual <- company_agg %>% select(-c(prev_predicted_ratio, new_pre
dicted_ratio)) %>% mutate(price_adjusted_ratio = actual_ratio, group = 'Actua
```

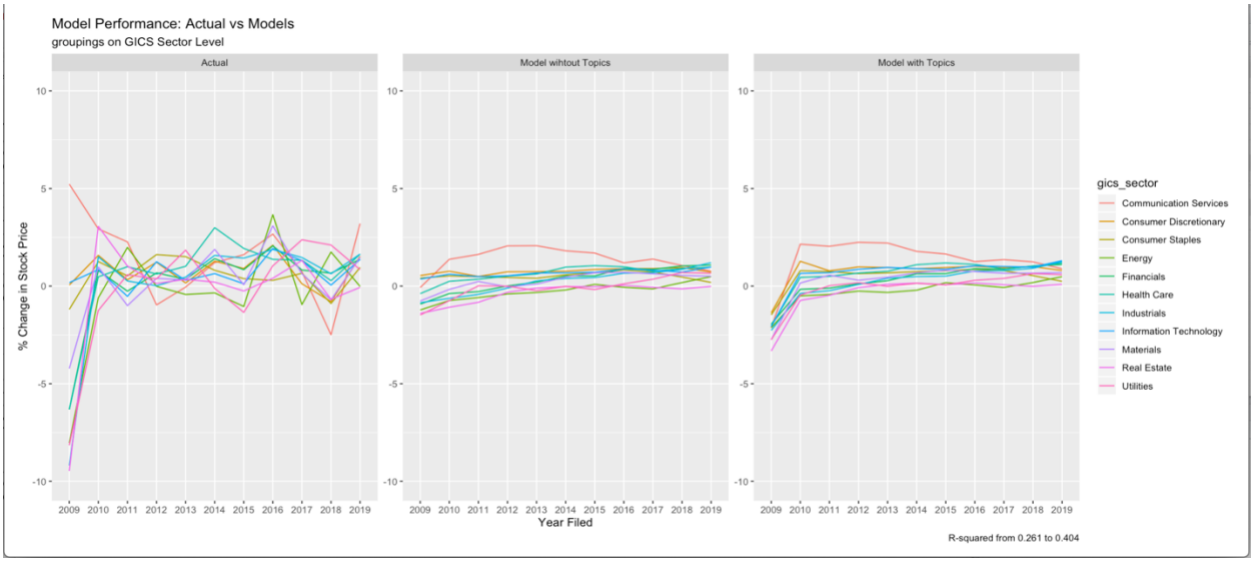
```
l') %>% select(-actual_ratio)
company_agg_predicted_prev <- company_agg %>% select(-c(actual_ratio, new_pre
dicted_ratio)) %>% mutate(price_adjusted_ratio = prev_predicted_ratio, group
= 'Model wihtout Topics') %>% select(-prev_predicted_ratio)
company_agg_predicted_new <- company_agg %>% select(-actual_ratio, prev_predi
cted_ratio) %>% mutate(price_adjusted_ratio = new_predicted_ratio, group = 'M
odel with Topics') %>% select(-new_predicted_ratio)

company_agg <- bind_rows(company_agg_actual, company_agg_predicted_prev, comp
any_agg_predicted_new)

# ----- Plot Actual vs Model
ggplot(company_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gic
s_sector)) + geom_line(aes(group = company_name)) + coord_cartesian(ylim=c(-5
0,50)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model P
erformance: Actual vs Models', subtitle = 'groupings on company Level',y = '%
Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ',
company_agg_rsquared_prev,' to ', company_agg_rsquared_new))

rm(company_agg, company_agg_rsquared_prev, company_agg_rsquared_new, company_
agg_actual, company_agg_predicted_prev, company_agg_predicted_new)
```



An improvement of 0.027 coefficient of determination on company level.

```
# ----- GICS Level
gics_agg <-  regression_data_full %>%
```

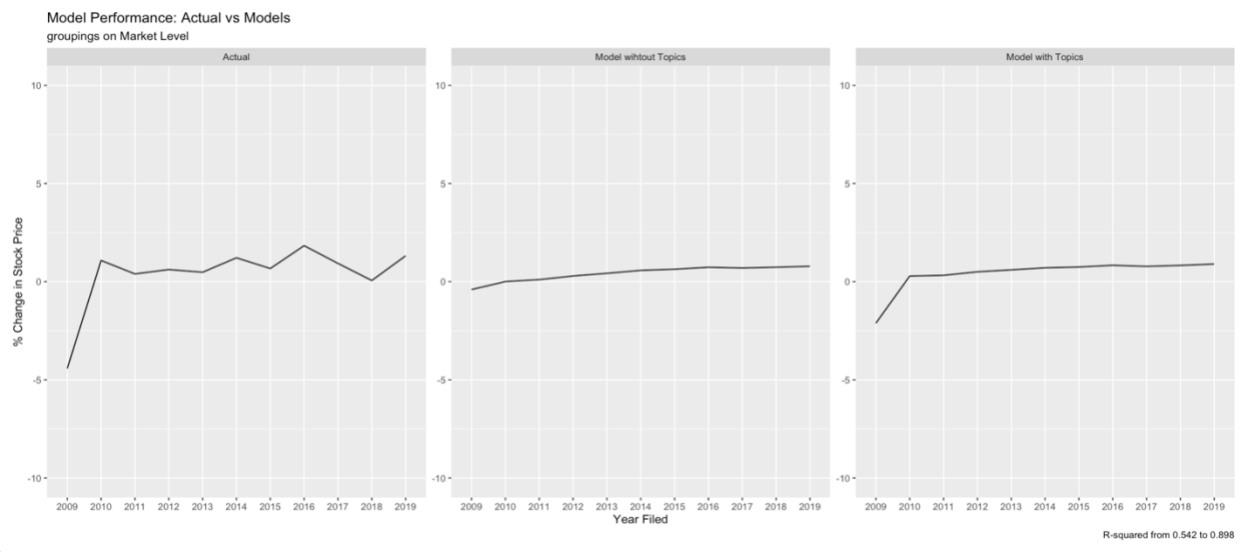
```
     group_by(year_filed, gics_sector) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              prev_predicted_ratio = mean(prev_predicted_ratio),
              new_predicted_ratio = mean(new_predicted_ratio))

gics_agg_rsquared_prev <- round(rsq(gics_agg$actual_ratio, gics_agg$prev_pred
icted_ratio), 3) # calculate R-squared between actual and predicted
gics_agg_rsquared_new <- round(rsq(gics_agg$actual_ratio, gics_agg$new_predic
ted_ratio), 3) # calculate R-squared between actual and predicted


gics_agg_actual <- gics_agg %>% select(-c(prev_predicted_ratio, new_predicted
_ratio)) %>% mutate(price_adjusted_ratio = actual_ratio, group = 'Actual') %>
% select(-actual_ratio)
gics_agg_predicted_prev <- gics_agg %>% select(-c(actual_ratio, new_predicted
_ratio)) %>% mutate(price_adjusted_ratio = prev_predicted_ratio, group = 'Mod
el wihtout Topics') %>% select(-prev_predicted_ratio)
gics_agg_predicted_new <- gics_agg %>% select(-actual_ratio, prev_predicted_r
atio) %>% mutate(price_adjusted_ratio = new_predicted_ratio, group = 'Model w
ith Topics') %>% select(-new_predicted_ratio)

gics_agg <- bind_rows(gics_agg_actual, gics_agg_predicted_prev, gics_agg_pred
icted_new)

# ----- Plot Actual vs Model
ggplot(gics_agg, aes(x = year_filed, y = price_adjusted_ratio, color = gics_s
ector)) + geom_line(aes(group = gics_sector)) + coord_cartesian(ylim=c(-10,10
)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model Perfo
rmance: Actual vs Models', subtitle = 'groupings on GICS Sector Level',y = '%
Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ',
gics_agg_rsquared_prev,' to ', gics_agg_rsquared_new))

rm(gics_agg, gics_agg_rsquared_prev, gics_agg_rsquared_new, gics_agg_actual,
gics_agg_predicted_prev, gics_agg_predicted_new)
```



An enhancement of 0.143 on sector level. As aggregation goes higher, it could be seen that the improvement is reflected in mainly year 2009.

```
# ----- Market Level
market_agg <-  regression_data_full %>%
     group_by(year_filed) %>%
    summarise(actual_ratio = mean(price_adjusted_ratio),
              prev_predicted_ratio = mean(prev_predicted_ratio),
              new_predicted_ratio = mean(new_predicted_ratio))

market_agg_rsquared_prev <- round(rsq(market_agg$actual_ratio, market_agg$pre
v_predicted_ratio), 3) # calculate R-squared between actual and predicted
market_agg_rsquared_new <- round(rsq(market_agg$actual_ratio, market_agg$new_
predicted_ratio), 3) # calculate R-squared between actual and predicted


market_agg_actual <- market_agg %>% select(-c(prev_predicted_ratio, new_predi
cted_ratio)) %>% mutate(price_adjusted_ratio = actual_ratio, group = 'Actual'
) %>% select(-actual_ratio)
market_agg_predicted_prev <- market_agg %>% select(-c(actual_ratio, new_predi
cted_ratio)) %>% mutate(price_adjusted_ratio = prev_predicted_ratio, group =
'Model wihtout Topics') %>% select(-prev_predicted_ratio)
market_agg_predicted_new <- market_agg %>% select(-actual_ratio, prev_predict
ed_ratio) %>% mutate(price_adjusted_ratio = new_predicted_ratio, group = 'Mod
el with Topics') %>% select(-new_predicted_ratio)

market_agg <- bind_rows(market_agg_actual, market_agg_predicted_prev, market_
agg_predicted_new)

# ----- Plot Actual vs Model
ggplot(market_agg, aes(x = year_filed, y = price_adjusted_ratio, group =1)) +
```

```
geom_line() + coord_cartesian(ylim=c(-10,10)) + facet_wrap(~group, ncol = 4, scales = "free") + labs(title ='Model Performance: Actual vs Models', subtitle = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared from ', market_agg_rsquared_prev,' to ', market_agg_rsquared_new))

rm(market_agg, market_agg_rsquared_prev, market_agg_rsquared_new, market_agg_actual, market_agg_predicted_prev, market_agg_predicted_new)
```



It becomes even more obvious on the market level that the model has improved its predictions for year 2009. Thus resulting to an improvement of 0.356, with a final predictability of almost 90%.

Conclusions

Topic features gathered from structural topic modelling has improved the exhaustive model introduced in part B by quite a lot. Sentiment analysis pales in comparison to how topic features have changed the predictability of the stock price % change.

Additionally, even in the most profound model we built for prediciting stock price % change by adding topic features, the volatility of the predicted outcome is no where near the actual trend. Therefore, we need to acknowledge that in order to increase the predictability, other variables outside textual features of financial reports that might relate to the instance such as behavioural economics features, should be captured to have more accurate prediction of stock price % change.

Reference

Brown, S. and Tucker, J., 2011. Large-Sample Evidence on Firms' Year-over-Year MD&A Modifications. *Journal of Accounting Research*, 49(2), pp.309-346.

Cabrera, N., 2014. *What You Should Know: Venezuelan SICAD II*. [online] ShipLilly. Available at: <https://www.shiplilly.com/blog/know-venezuelan-sicad-ii/>.

Garrett, T., 2003. Aggregated versus disaggregated data in regression analysis: implications for inference. *Economics Letters*, 81(1), pp.61-65.

Kenton, W., 2018. *Financial Stability Plan (FSP)*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/f/financial-stability-plan-fsp.asp>.

Roberts, M., Stewart, B. and Tingley, D., 2019. stm: An R Package for Structural Topic Models. *Journal of Statistical Software*, 91(2).