edgar

Gibran Makyanie

03/04/2020

# ETL Initiation

## 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))

## 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 with 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)

## Import Stop Words from Loughran & McDonald’s

# ----- 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, local\_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

## Process MD&A chapter 10K reports

conn <- dbConnect(RSQLite::SQLite(), "edgar.db")  
  
the\_index <- dbGetQuery(conn, 'SELECT accession\_number FROM master\_index WHERE form\_type = "10-K" ORDER BY date\_filed')  
  
split\_size <- 50  
split\_the\_index <- split(the\_index$accession\_number, ceiling(seq\_along(the\_index$accession\_number)/split\_size))  
  
rm(the\_index)  
  
for (s in 1:length(split\_the\_index)) {  
file\_pattern <- paste0(split\_the\_index[[s]], '.txt')  
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$company\_name, " ", n= nchar(text\_transformed$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 master\_index 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 10Q reports

the\_index <- dbGetQuery(conn, 'SELECT cik, accession\_number FROM master\_index WHERE form\_type = "10-Q" ')  
z = 0  
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 <- regmatches(doc,gregexpr("(?<=notes).\*",doc,perl=TRUE))[[1]]}   
 if(length(doc\_begin) > 0) {doc\_begin <- doc\_begin} else {doc\_begin <- regmatches(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 and results of operations",  
 "managements discussion and analysis of financial condition and 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\_index 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\_10q.accession\_number = master\_index.accession\_number   
 WHERE cleaned\_10q.accession\_number = "',local\_df$accession\_number[1],'"'))  
   
 print(paste(temp\_report$form\_type, temp\_report$quarter, 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)  
 }  
z = z + 1  
print(paste(z, "out of 499"))  
  
}

# Text Cleaning

## 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\_index 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)  
  
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;') # add cleaned\_text to master\_index  
dbExecute(conn, 'ALTER TABLE master\_index ADD COLUMN cleaned\_noun TEXT;') # add clean noun column to master\_index

## Retokenise 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.accession\_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 = '",accession\_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\_text ,"', cleaned\_noun = '",cleaned\_noun ,"' WHERE accession\_number = '",accession\_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\_filed) %>%  
 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()

# ----- 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', '2018,2019')  
  
for (era in 1:length(financial\_era)) {  
  
 gics\_level <- dbGetQuery(conn, paste0('SELECT gics\_sector, cleaned\_noun FROM 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 = FALSE) +  
 labs(x = NULL, y = "tf-idf", title = paste("Important Terms", financial\_era[era]), subtitle = "grouped by GICS sector") +  
   
 facet\_wrap(~gics\_sector, ncol = 4, scales = "free") + coord\_flip()  
  
}