Code Appendix

```r
## Word2Vec
library(tidyverse)
library(quanteda)
library(word2vec)
library(parallel)

## Parallel
sock<-makePSOCKcluster(3L) ##3 Procesadores!
clusterSetRNGStream(cl=sock, 10)
clusterEvalQ(sock,{library(quanteda) ; library(tidyverse)})

## Read Data
setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
data <- read_csv("Data/AllData.csv") %>%
  separate(Date, into = c("Date", "Author"), sep = "\\|") %>%
  mutate(Date = ymd_hms(as.character(strptime(str_squish(Date), format = "%B %d, %Y at %H:%M"))),
         Author = str_squish(Author)) %>%
  filter(URL != "https://oilprice.com/Latest-Energy-News/World-News/Russias-Lukoil-Ships-Arctic-Oil-To-C

## Example
ex <- data$Text

## Basic Cleaner Function
easy_parse <- function(piece){
  x <- str_remove(piece, "[Oo]il[Pp]rice\\.com.+")
  x <- str_replace_all(x, '[\\n""]', "")
  writer <- str_extract(x, "By [A-Z][a-z]+ [A-Z][a-z]+ for")
  writer <- ifelse(is.na(writer), "NONE", writer)
  x <- str_remove(x, writer) %>% str_to_lower()
  writer <-str_remove_all(writer, "By|for") %>% str_squish()
  bow <- quanteda::tokens(x, remove_punct = T, remove_symbols = T, remove_numbers =T)
  out <- str_c(bow[[1]], collapse = " ") %>% str_replace_all("[\\.]", " ")
  out <- str_replace_all(out, "u s", "u.s")
  list("Text"= out, "Editor"=writer)
}

## Export
clusterExport(sock, ls())

## Doc clean
system.time({new_docs <- parLapply(sock, ex, easy_parse)})
new_text <- unlist(lapply(new_docs, `[[`, 1))

##
data$Text <- new_text
data <- data %>% distinct()
```

```r
write_rds(data , "DataOil.Rdata")

## Word2Vec
model <- word2vec(x = new_text, dim = 50, iter = 20)
emb   <- as.matrix(model)
head(emb)

## Vocabulary
vocab <- summary(model, type = "vocabulary")
vocab

## Relations
emb_p <- predict(model, c("reservoir"), type = "nearest")
emb_p

## Save the model to hard disk
path <- "mymodel.bin"
write.word2vec(model, file = "C:/Users/dordo/Documents/Daniel/LSE/MY 459/ProyectoModelWV.bin")

##----------------------------------------------------------------------------------------------##
##----------------------------------------------------------------------------------------------##

library(tidyverse)
library(quanteda)
setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")

## Read RData
data <- read_rds("DataOil.Rdata")

## Filter only relevant articles
check_data <- data %>% filter(year(Date) > 2018, year(Date) < 2021)

## Count by Author
check_data %>% distinct(Author)

##-----------------------------------------------------------------------##
## Attention Plot
tot_date <- tibble("Date"=ymd(seq.Date(from = as.Date("2019/01/01"),
                                       to = as.Date("2020/12/31"), "days")))

sum_data <- check_data %>% mutate(Date = floor_date(Date, unit = "day")) %>%
  group_by(Date) %>% summarise("Count"=n())

sum_data <- tot_date %>% left_join(sum_data) %>%
  mutate(Missing = ifelse(is.na(Count), 1L , 0L))

sum_data %>% ggplot(aes(x=Date, y= Count)) + geom_line(color = "red") +
  ggtitle("Daily Count of Articles") + theme_minimal()

colrs <- c("#00FF00", "#FF0000")
sum_data %>% mutate(Year = as.factor(year(Date))) %>%
  ggplot(aes(x=Count, fill = Year)) + geom_bar() + facet_wrap(~Year) +
  xlab("# of articles") + theme_minimal() +
  scale_fill_manual(values = colrs)

##----------------------------------------------------------------------##
## Number of tokens by news
corp <- corpus(check_data %>% mutate(Year=year(Date)), text_field = "Text")
tks <- tokens(corp, remove_punct = T, remove_symbols = T, remove_numbers = T)
tks <- tokens_remove(tks, c(stopwords("en"), "oil", "curde", "said", "u."))
dfmat <- dfm(tks, stem = T, groups = "Year")
dfmat

## WordCloud
quanteda::textplot_wordcloud(dfm_subset(dfmat, Year == 2019), max_words = 100, color = "darkgreen")
quanteda::textplot_wordcloud(dfm_subset(dfmat, Year == 2020), max_words = 100, color = "red")


##-----------------------------------------------------------------------##
## Summary
sum_tks <- textstat_summary(tks) %>% select(tokens, types) %>%
  mutate(Id=seq_along(tokens)) %>%
  pivot_longer(cols = c(tokens, types))

ggplot(sum_tks %>% filter(name == "tokens"), aes(x=log(value))) +
  geom_histogram(fill = "red") +
  theme_minimal()

statk <- textstat_keyness(dfmat, target = "2020")

data_key <- bind_rows(statk %>% slice_max(chi2, n = 20),
                      statk %>% slice_min(chi2, n = 20) %>% arrange(desc(chi2))) %>%
  as_tibble() %>%
  select(feature, chi2) %>%
  mutate(id = seq_along(feature),
         Year = ifelse(id < 21, "2020", "2019"))

##
ggplot(data_key, aes(x=id, y=chi2, fill = Year)) + geom_col() +
  geom_text(aes(label=feature),  angle = 45, size = 3,
            position = position_nudge(x=0, y = c(rep(20, 20), rep(-20, 20)))) +
  facet_wrap(~Year, scales = "free") + theme_minimal() +
  scale_fill_manual(values = colrs) + theme(axis.title.x=element_blank(),
                                            axis.text.x=element_blank(),
                                            axis.ticks.x=element_blank())

#textstat_lexdiv(tks)

##--------------------------------------------------------------------------------------------##
##--------------------------------------------------------------------------------------------##

setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
library(tidyverse)
library(readxl)

## Read Polarity Scores
```

```r
pol_scores <- read_csv("Data/PolScores.csv")

## Read controls
cntrls <- read_excel("Data/Controls.xlsx") %>%
  mutate(US10=ifelse(US10 == 0, NA, US10),
         Vix=ifelse(Vix == 0, NA, Vix),
         Gold = ifelse(Gold==0, NA, Gold)) %>%
  mutate(US10 = c(NA, diff(US10)),
         Gold = c(NA, diff(log(Gold))),
         Vix = c(NA, diff(Vix)))


## Read Oil Scores Forwards
oil_fws <- read_excel("Data/OilPrices.xlsx") %>%
  mutate(Price = ifelse(Price > 0, Price, 0.01)) %>%
  fill(Price, .direction = "down") %>%
  mutate(DLPrice = c(NA,diff(log(Price))))

# oil <- read_excel("Data/OilPrices.xlsx", sheet=2)  %>%
#   mutate(Price = ifelse(Price > 0, Price, 0)) %>%
#   fill(Price, .direction = "down") %>%
#   mutate(DLPrice = c(NA,diff(log(Price))))

#data_oil <- oil_fws %>% full_join(oil, by = "Date") %>% arrange(Date)
#lm(data_oil$Price.x ~ data_oil$Price.y) %>% summary()

##---------------------------------------------------------------------##
## Consolidate Dataset

data <- pol_scores %>% left_join(oil_fws) %>%
  filter(!is.na(Price)) %>%
  mutate(DumCovid = as.numeric(year(Date) == 2020 & month(Date) > 2),
         DLPriceDir = ifelse(sign(DLPrice) == 1, 1, 0),
         McDonald = ifelse(!is.na(McDonald), McDonald, 0),
         Outlier = ifelse(Date == ymd("2020-04-20"), 1, 0),
         Outlier = ifelse(Date == ymd("2020-04-21"), -1, Outlier)) %>%
  left_join(cntrls)

#"2020-04-21""2020-04-20"
## Modelling outliers
model <- forecast::auto.arima(ts(data$DLPrice, freq=5), xreg = data$Outlier)
data_use <- bind_cols(data, "Ehat"=residuals(model)) %>%
  filter(year(Date) > 2018 & year(Date) < 2021)


##--------------------------------------------------------------------##
## Basic Cross Correlation Analysis
ccf(data_use$DLPrice, data_use$McDonald, na.action = na.omit)


write_csv(data_use, "data/DataReg.csv")

##--------------------------------------------------------------------------------------------##
```

```r
##----------------------------------------------------------------------------------------------##

setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
library(tidyverse)

## Read McDonald
check_words <- str_to_lower(read_csv("Data/CheckWordsMC.csv")[[1]])
pos_modifiers <- str_to_lower(read_csv("Data/PositiveModifiers.csv")[[1]])
neg_modifiers <- str_to_lower(read_csv("Data/NegativeModifiers.csv")[[1]])

## Load RData
data <- read_rds("DataOil.RData")

##---------------------------------------------------------------------------------------##
## Function
search_txt <- function(x, word_pool, pos, neg){
  tks <- tokens(x, remove_numbers = T, remove_symbols = T,
                remove_punct = T)
  tks <- tokens_remove(tks, stopwords("en"))
  search_wrds <- function(tks, x, pos, neg){
    aux <- kwic(tks, x)
    if(length(aux[[1]]) > 0){
      context <- c(str_split(aux$pre, " ", simplify = T), str_split(aux$post, " ", simplify = T))
      posc <- sum(pos %in% context)
      negc <- sum(neg %in% context)
      c(posc, negc)
    } else{
      c(0, 0)
    }
  }
  structure(colSums(do.call(rbind, lapply(word_pool, search_wrds, tks = tks, pos = pos, neg = neg))),
            names = c("Positive", "Negative"))
}

##system.time({search_txt(data$Text[[1]], check_words, pos_modifiers, neg_modifiers)})
##search_txt(data$Text[[10]], check_words, pos_modifiers, neg_modifiers)
res <- lapply(data$Text, search_txt, check_words, pos_modifiers, neg_modifiers)
mat <- do.call(rbind, res)
out_data <- bind_cols(data, "Positive"=mat[,1], "Negative"=mat[,2])
colnames(out_data)
write_csv(out_data, "Data/McDonaldOriginal.csv")

##----------------------------------------------------------------------------------------------##
##----------------------------------------------------------------------------------------------##

setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
library(tidyverse)
library(quanteda)

##
word_list <- map(c("NegativeModifiers.csv", "NegativeWordsMC.csv",
                   "PositiveModifiers.csv", "PositiveWordsMC.csv"),
                 ~as.character(read_csv(str_c("Data/", .x))[[1]]))
```

```r
names(word_list) <- c("NegativeModifiers", "NegativeWords", "PositiveModifiers",
                      "PositiveWords")

## Dictionary
mc_dict <- quanteda::dictionary(word_list)

## Read data
data <- read_rds("DataOil.Rdata")

## Convert to Corpus
data_c <- corpus(data, docid_field = "Title", text_field = "Text")

## Apply Mcdonald Dictionary
scores <- dfm(data_c, dictionary = mc_dict)

## By sentiment
neg_scores <- dfm_keep(scores, "Negative", valuetype = "regex") %>% rowSums()
pos_scores <- dfm_keep(scores, "Positive", valuetype = "regex") %>% rowSums()

## Total
total <- (pos_scores - neg_scores)/(neg_scores + pos_scores)

## Export
out <- tibble("Title"=names(total), "Score"=total)

write_csv(out, "data/McDonaldPolarities.csv")

##---------------------------------------------------------------------------------------------##
##---------------------------------------------------------------------------------------------##
setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
library(tidyverse)
library(quanteda)
library(lubridate)
##---------------------------------------------------------------------------------------##
## Read Polarity Scores

## Semantic Axis
data_sem <- read_csv("data/SemanticAxisPolarities.csv") %>% rename(SemanticAxis = Sim)
data_semMc <- read_csv("data/SemanticAxisPolaritiesMc.csv") %>% rename(SemanticAxisMc = Sim)

## Sentiprop
data_senti <- read_csv("data/SentiPropPolarities.csv") %>% rename(Sentiprop = polarity)
data_sentiMc <- read_csv("data/SentiPropPolaritiesMc.csv") %>% rename(SentipropMc = polarity)

## McDonald
#data_mcdof <- read_csv("data/McDonaldPolarities.csv") %>% rename(McDonald = Score)
#data_mcdo <- read_csv("data/McDonaldOriginal.csv") %>% rename(McDonald = Score)

## Join
scaler <- function(x){
  #(x-min(x))/(max(x)-min(x))
  as.numeric(scale(x))
}
```

```
pol_data <- inner_join(inner_join(data_sem, data_semMc),
                       inner_join(data_senti, data_sentiMc),
                       by =c("Word"="words")) %>%
  mutate(across(!c(Word), scaler))

rm(list=ls(pattern="^data"))

##--------------------------------------------------------------------------------------------##
## Exploratory plots
#z <- "Sentiprop"
get_keyw <- function(z, pol_data){
  bind_rows(pol_data %>% select(Word, .data[[z]]) %>% slice_max(.data[[z]], n = 10),
            pol_data %>% select(Word, .data[[z]])  %>% slice_min(.data[[z]], n = 10)) %>%
    arrange(desc(.data [[z]]))
}

nams <- c("Sentiprop", "SentipropMc", "SemanticAxis", "SemanticAxisMc")
keyword <- lapply(nams, get_keyw,
                  pol_data = pol_data)

tab <- do.call(cbind, map(keyword, 1))
colnames(tab) <- nams
xtable::xtable(tab[1:10,])


## Incoherency
pair_comp <- function(x, y, pol_data){
  use_data <- pol_data %>% select(Word, .data[[x]], .data[[y]])
  tab <- use_data %>%
    mutate(across(!Word, sign)) %>%
    rowwise() %>%
    mutate(p = prod(c_across(!Word))) %>%
    count(p)
  list("Incoherency"=tab[[2]][1]/(tab[[2]][1]+tab[[2]][2]),
       "Cor"=cor(use_data[[2]], use_data[[3]]))
}

incs <- c(pair_comp("Sentiprop", "SentipropMc", pol_data)$Incoherency,
          pair_comp("Sentiprop", "SemanticAxis", pol_data)$Incoherency,
          pair_comp("Sentiprop", "SemanticAxisMc", pol_data)$Incoherency,
          pair_comp("SentipropMc", "SemanticAxis", pol_data)$Incoherency,
          pair_comp("SentipropMc", "SemanticAxisMc", pol_data)$Incoherency,
          pair_comp("SemanticAxis", "SemanticAxisMc", pol_data)$Incoherency)

incs_mat <- matrix(NA, 4, 4)
incs_mat[upper.tri(incs_mat)] <- incs
xtable::xtable(incs_mat)

pair_comp("SemanticAxisMc", "SemanticAxis", pol_data = pol_data)

library(GGally)
ggpairs(pol_data %>% select(-Word))
```

```r
##---------------------------------------------------------------------------------------##
## Read news data
data <- read_rds("DataOil.Rdata")

## Quanteda
datos <- corpus(data, docid_field = "Title", text_field = "Text")

dfm_oil <- dfm(datos, select = pol_data$Word) %>%
  dfm_weight(scheme = "prop")

weight_fun <- function(x, data){
  # weight by the weights
  dfm_c <- data  %>%
    dfm_weight(weights = structure(pol_data[[x]], names=pol_data$Word))
  sents <- rowSums(dfm_c)
  out <- tibble(names(sents), sents)
  colnames(out) <- c("Title", x)
  out
}

## Use function above
vars <- c("Sentiprop", "SentipropMc", "SemanticAxis", "SemanticAxisMc")
res <- lapply(vars, weight_fun, data = dfm_oil)

## Reduce to inner join
data_score <- data %>% select(Date, Title, Author) %>%
  left_join(reduce(res, left_join))

##----------------------------------------------------------------------------------##
## Append Mcdonald info
data_mcd <- left_join(read_csv("Data/McDonaldPolarities.csv") %>%
                        mutate(Score = ifelse(is.na(Score), 0, Score)),
                      read_csv("Data/McDonaldOriginal.csv") %>%
                        select(Title, Positive, Negative) %>%
                        mutate(Total = (Positive - Negative)/(Negative + Positive)) %>%
                        mutate(Total = ifelse(is.na(Total), 0, Total)) %>%
                        select(Title, Total)) %>%
  rename(McDonaldF = Score, McDonald = Total)

data_score <- data_score %>% left_join(data_mcd)

##-----------------------------------------------------------------------------------##
## Aggregate
pol_scores <- data_score %>% mutate(Date=round_date(Date, "day")) %>%
  group_by(Date) %>%
  summarise(across(where(is.numeric), sum),
            N = n()) %>%
  select(Date, everything()) %>%
  arrange(Date)

## Save
#write_csv(pol_scores, "data/PolScores.csv")
```

```
## Plots

## By week
res <- pol_scores %>%
  mutate(Date = round_date(Date, unit = "week")) %>%
  group_by(Date) %>%
  summarise(across(everything(),sum)) %>%
  filter(year(Date)>2018, year(Date)<2021)

res %>% select(-N) %>%
  mutate(across(!Date, ~Get_trend(.x, type = "Henderson", m=4)),
         Date = as.Date(Date)) %>%
  pivot_longer(Sentiprop:McDonald, names_to = "Method") %>%
  ggplot(aes(x=Date, y=value, color = Method)) + geom_line() +
  scale_x_date(date_labels = "%m-%y") +
  facet_wrap(~Method, scales = "free", nrow = 3) +
  theme_minimal()

# ggplot(pol_scores %>%
#          pivot_longer(Sentiprop:SemanticAxisMc, names_to = "Method", values_to = "Score"),
#        aes(x=Date, y=Score, col = Method)) + geom_line() + facet_wrap(~Method)
#
# ggplot(pol_scores, aes(x=SemanticAxisMc, y=SemanticAxis)) + geom_point()
# ggplot(pol_scores, aes(x=Sentiprop, y=SentipropMc)) + geom_point()

##---------------------------------------------------------------------------------------------##
##---------------------------------------------------------------------------------------------##

library(tidyverse)
library(word2vec)
setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
model <- word2vec::read.word2vec("ModelWV.bin")

## record vocabulary
vocab <- summary(model)

#a <- as.matrix(model)
#b <- data.frame("name"=rownames(a), a)
##-------------------------------------------------------------------------------##
## Check for good seeds
# root_seed <- c("succes", "excel", "^profit", "benefic", "improv",
#                 "positiv", "^gain")

## A la McDonald
root_seed <- c("^discov", "glut", "^overpro", "^oversup", "recess",
               "^repair", "^surpl")

## Matches in the vocabulary
root_words <- unlist(lapply(root_seed, str_subset, string = vocab))
#write_csv(as_tibble(root_words), "data/rootGoodMc.csv")

## Embedding for root words
good_embedding <- predict(model, root_words, "embedding")
```

```
## Check: Exploratory Factor Analysis of Data Matrices With More Variables Than Observations
good <- colMeans(good_embedding)
hist(good_embedding[,3])

##-------------------------------------------------------------------------------##
## Check for bad seeds
# root_seed_bad <- c("^los", "volati", "wron", "damag", "^bad", "litiga",
#   "fail", "negat", "downg", "lock[^c]", "down[swt].",
#   "^[^ld].+down$")

root_seed_bad <- c("attack", "bomb", "^closur", "^concer", "confli",
                   "delay", "^dispu", "^disrup", "^explos", "fire",
                   "^hurri", "instab", "outa", "probl", "recover",
                   "sabo", "shorta", "storm", "strike", "turm",
                   "unres", "withd")

# new_roots <- read_csv("Data/NegativeWordsMC.csv")
# new_roots %>% pull() %>% str_to_lower() %>% view()

root_words_bad <- unlist(lapply(root_seed_bad, str_subset, string = vocab))
#write_csv(as_tibble(root_words_bad), "data/rootBadMc.csv")

## Embedding for root words
bad_embedding <- predict(model, root_words_bad, "embedding")
bad <- colMeans(bad_embedding)
hist(bad_embedding[,3])

##----------------------------------------------------------------------------------##
## Calculate semantic axis
sem_axis <- good-bad
dot_p <- t(sem_axis %*% t(as.matrix(model)))/50
new_dict <- tibble("Word"=rownames(dot_p), "Sim"=dot_p[,1]) %>% arrange(desc(Sim))

write_csv(new_dict, "data/SemanticAxisPolaritiesMc.csv")

## Final
predict(model, "success", type = "nearest")

##---------------------------------------------------------------------------------------------##
##---------------------------------------------------------------------------------------------##

setwd("C:/Users/dordo/Documents/Daniel/LSE/MY 459/Proyecto")
library(tidyverse)
library(caret)
library(glmnet)

## Read csv
data <- read_csv("data/DataReg.csv")

## Auxiliary function for lags
get_lags <- function(x, y, max_l){
  x_use <- x %>% select(2)
  aux <- do.call(cbind,map(max_l:0,~lag(x_use,.x)))
```

```r
  colnames(aux) <- str_c(y, str_c("_Lag", max_l:0))
  cbind(aux, "Date"=x[[1]])
}

go_pred <- function(z, data, max_lag = 1){
  ## Calculate Lags
  test <- data %>% select(Date, Ehat, .data[[z]], N, US10, Gold, Vix) %>%
    pivot_longer(Ehat:Vix) %>%
    group_nest(name)

  ## Generate lags
  test_df <- test %>%
    mutate(data = map2(data, test$name, get_lags, max_l = max_lag)) %>%
    select(data) %>%
    pull() %>%
    reduce(left_join) %>%
    as_tibble() %>%
    select(Date, everything())

  ## Find missings and filter out
  nas <- rowwise(test_df) %>% summarise(s = sum(is.na(c_across(!Date))))
  X_mat <- as.matrix(test_df %>% select(-Date) %>% filter(nas == 0))

  set.seed(5)
  test_ind <- sample(1:nrow(X_mat), 50)

  ## Calculate model
  id <- (max_lag + 1)

  ## GLMNET
  # model <- cv.glmnet(X_mat[-test_ind, id], x = X_mat[-test_ind,-id])
  # yhat <- predict(model, X_mat[test_ind, -id], s = "lambda.min")

  ## Using leaps
  model <- regsubsets(x=X_mat[-test_ind,-id], y=X_mat[-test_ind, id])
  summ <- summary(model)
  use_reg<-summ$which[which.min(summ$bic),]
  new_mat <- cbind("Ehat"=X_mat[, id],X_mat[, -id][, use_reg[-1]])
  model_2 <- lm(Ehat ~ ., data = as.data.frame(new_mat[-test_ind,]))
  yhat <- predict(model_2, as.data.frame(new_mat[test_ind,]))

  rmse_t <- sqrt(mean((X_mat[test_ind, id] - yhat)^2))
  list(model_2, rmse_t, X_mat)
}

z <- c("Sentiprop", "SentipropMc", "SemanticAxis",
       "SemanticAxisMc", "McDonaldF", "McDonald")

res <- lapply(z, go_pred, data = data, max_lag = 2)
lapply(res, `[[`, 1)
lapply(res, `[[`, 2)
```