CBE2

Lee, Woo Chan

10/31/2021

```
library(quanteda)
library(quanteda.textstats)
library(tidyverse)
library(cluster)
library(factoextra)
library(cmu.textstat)
library(stringr)
library(anytime)
library(stringr)
library(dendextend)
library(ggdendro)
library(janitor)
library(data.table)
```

lab 11: vector embedding in plot (2014~2019 tsla comparison)

```
# Read main and meta data
twt <- read_csv("/Users/lee14257/Development/CMU/Text Analysis/Project/archive/Tweet.csv")
```

```
## Rows: 3717964 Columns: 7
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): writer, body
## dbl (5): tweet_id, post_date, comment_num, retweet_num, like_num
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#twt <- read_csv("/Users/lee14257/Development/CMU/Text Analysis/Project/archive/Tweet_sample.csv")
twt_bytickers <- fread("/Users/lee14257/Development/CMU/Text Analysis/Project/CBE2/twt_bytickersymbol.cs
meta <- read_csv("/Users/lee14257/Development/CMU/Text Analysis/Project/archive/Company_Tweet.csv")
```

```
## Rows: 4336445 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ticker_symbol
## dbl (1): tweet_id
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Change "GOOGL" ticker to "GOOG" for consistency
meta$ticker_symbol[meta$ticker_symbol == "GOOGL"] <- "GOOG"

# Merge ticker_symbol meta data to twt
twt <- merge(x = twt, y = meta, by = "tweet_id", all.x = TRUE)

# Drop NA values
twt <- twt[!is.na(twt$ticker_symbol), ] %>%
  mutate(post_date = format(anytime(post_date), "%Y-%m"))
head(twt)
```

```
##       tweet_id          writer post_date
## 1 5.504415e+17 VisualStockRSRC   2014-12
## 2 5.504417e+17     KeralaGuy77   2014-12
## 3 5.504417e+17     DozenStocks   2014-12
## 4 5.504430e+17    ShowDreamCar   2014-12
## 5 5.504438e+17    i_Know_First   2014-12
## 6 5.504438e+17    i_Know_First   2014-12
##
## 1                           lx21 made $10,008  on $AAPL -Check it out! http://profit.ly/1MnD8s?aff=2
## 2                                                  Insanity of today weirdo massive selling. $aa
## 3                                S&P100 #Stocks Performance $HD $LOW $SBUX $TGT $DVN $IBM $AMZN $F
## 4 $GM $TSLA: Volkswagen Pushes 2014 Record Recall Tally Higher https://pic.twitter.com/WIIc1lW7hW @P
## 5                                                  Swing Trading: Up To 8.91% Return In 14 Days
## 6                                                  Swing Trading: Up To 8.91% Return In 14 Days
##   comment_num retweet_num like_num ticker_symbol
## 1           0           0        1          AAPL
## 2           0           0        0          AAPL
## 3           0           0        0          AMZN
## 4           0           0        1          TSLA
## 5           0           0        1          AAPL
## 6           0           0        1          TSLA
```

Create twitter token table (company tickers)

```r
# This code was pre-run and stored into a CSV file so that we didn't have to
# run it again (took a long time to run)
# twt_tkn <- twt %>%
#   dplyr::select(ticker_symbol, body) %>%
#   group_by(ticker_symbol) %>%
#   summarise(text = paste(body, collapse=" ")) %>%
#   mutate(
#     doc_id = ticker_symbol,
#     text = tolower(text)
#     )
names(twt_bytickers)[1] <- 'doc_id'
names(twt_bytickers)[2] <- 'text'
twt_bytickers$text <- tolower(twt_bytickers$text)
```

```r
# Create token (by company tickers)
twt_tkn <- twt_bytickers %>%
```

```
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE, remove_punct = TRUE,
         remove_symbols = TRUE, remove_url=TRUE) %>%
  tokens_remove(c('\\$[a-z0-9]+', '\\#[a-z0-9]+', '[0-9]+\\%', '\\@[a-z0-9]+'),
                valuetype='regex') %>%
  tokens_remove(c(stopwords("english"), "apple", "appl", "aapl", "apple's",
                  "amazon's", "amzn", "amazon", "google's", "google", "googl",
                  "goog", "microsoft", "microsoft's", "msft", "tsla", "tesla",
                  "tesla's"))
```

```
# Create docvar for the tokens
doc_ticker <- names(twt_tkn) %>%
  data.frame(ticker = .)

docvars(twt_tkn) = doc_ticker
```

```
# Create dfm using tokens
twt_dfm <- twt_tkn %>% dfm()
```

Corpus Composition Table

```
# Corpus composition table
twt_comp <- ntoken(twt_dfm) %>%
  data.frame(Tokens = .) %>%
  rownames_to_column("Company Ticker") %>%
  janitor::adorn_totals("row")
```

```
# Corpus composition table
kableExtra::kbl(twt_comp, caption = "Composition of the twitter corpus",
                booktabs = T, linesep = "", format.args = list(big.mark = ",")) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic() %>%
  kableExtra::row_spec(5, hline_after = TRUE) %>%
  kableExtra::row_spec(6, bold=T)
```

Table 1: Composition of the twitter corpus

Company Ticker	Tokens
AAPL	10,816,314
AMZN	5,627,103
GOOG	5,184,923
MSFT	2,886,835
TSLA	10,846,623
Total	**35,361,798**

Keyness Tables

```
# Create keyness tables for the 5 company tickers
aapl_kw <- textstat_keyness(twt_dfm, docvars(twt_dfm, "ticker") == "AAPL",
                            measure = "lr") %>%
  as_tibble() %>% dplyr::select(feature, G2) %>% rename(LL = G2, Token = feature)
amzn_kw <- textstat_keyness(twt_dfm, docvars(twt_dfm, "ticker") == "AMZN",
                            measure = "lr") %>%
  as_tibble() %>% dplyr::select(feature, G2) %>% rename(LL = G2, Token = feature)
goog_kw <- textstat_keyness(twt_dfm, docvars(twt_dfm, "ticker") == "GOOG",
                            measure = "lr") %>%
  as_tibble() %>% dplyr::select(feature, G2) %>% rename(LL = G2, Token = feature)
msft_kw <- textstat_keyness(twt_dfm, docvars(twt_dfm, "ticker") == "MSFT",
                            measure = "lr") %>%
  as_tibble() %>% dplyr::select(feature, G2) %>% rename(LL = G2, Token = feature)
tsla_kw <- textstat_keyness(twt_dfm, docvars(twt_dfm, "ticker") == "TSLA",
                            measure = "lr") %>%
  as_tibble() %>% dplyr::select(feature, G2) %>% rename(LL = G2, Token = feature)
```

Table 2: Tokens with the highest keyness values in the AAPL ticker compared to the rest

Token	LL
iphone	191672.77
free.	98696.51
join	86281.90
use	76589.93
sure	75171.72
link.	51450.10
make	47238.19

Table 3: Tokens with the highest keyness values in the AMZN ticker compared to the rest

Token	LL
prime	48210.12
bezos	20352.58
aws	13991.38
alexa	13807.33
com	12685.23
foods	11804.44
jeff	11645.37

Table 4: Tokens with the highest keyness values in the MSFT ticker compared to the rest

Token	LL
windows	117398.13
more:	84994.70
read	81569.35
xbox	41470.34
surface	35569.67
corporation	26130.03
azure	17460.60

Table 5: Tokens with the highest keyness values in the GOOG ticker compared to the rest

Token	LL
alphabet	124994.69
class	39869.03
inc.	28956.59
inc	23332.10
android	22689.58
pixel	20712.26
c	18851.51

Table 6: Tokens with the highest keyness values in the TSLA ticker compared to the rest

Token	LL
model	104800.61
elon	96136.50
musk	90075.91
cars	37737.84
car	34568.03
production	21221.23
ev	18638.86

Load Docuscope Dictionary

```
# Add docuscope dictionary
ds_dict <- dictionary(file = "/Users/lee14257/Development/CMU/Text Analysis/Project/ds_dict.yml")
```

Create twitter token table (for time series)

```
# Preprocessing twitter token table for time series analysis
twt_time_tkn <- twt %>%
  dplyr::select(ticker_symbol, post_date, body) %>%
  group_by(ticker_symbol, post_date) %>%
  summarise(text = paste(body, collapse=" ")) %>%
  mutate(
    doc_id = paste0(ticker_symbol, "_", post_date),
    text = tolower(text)
    ) %>%
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE, remove_punct = TRUE,
         remove_symbols = TRUE, remove_url=TRUE) %>%
  tokens_remove(c(stopwords("english"), "apple", "appl", "aapl", "apple's",
                  "amazon's", "amzn", "amazon", "google's", "google", "googl",
                  "goog", "microsoft", "microsoft's", "msft", "tsla", "tesla",
                  "tesla's")) %>%
  tokens_remove(c('\\$[a-z]+', '\\#[a-z]+', '[0-9]+\\%', '\\@[a-z0-9]+'),
                valuetype='regex')
```

```
## `summarise()` has grouped output by 'ticker_symbol'. You can override using the `.groups` argument.
```

Apply docuscope word tagging

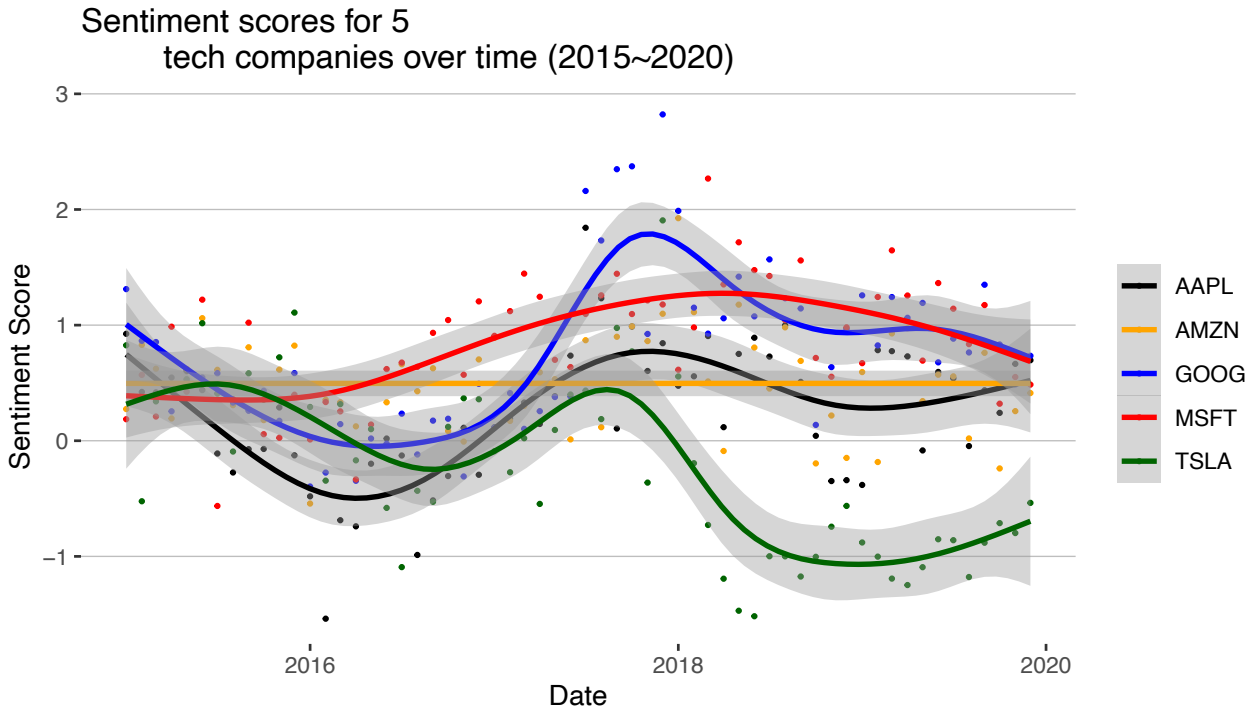
```
# Tag the tokens using docuscope
ds_counts <- twt_time_tkn %>%
  tokens_lookup(dictionary = ds_dict, levels = 1, valuetype = "fixed") %>%
  dfm() %>%
  convert(to = "data.frame") %>%
  as_tibble() %>%
  mutate(
    # Add sentiment score
    sentiment_score = positive - negative
  )
```

```
# Normalize the counts
tot_counts <- quanteda::ntoken(twt_time_tkn) %>%
  data.frame(tot_counts = .) %>%
  tibble::rownames_to_column("doc_id") %>%
  dplyr::as_tibble()

ds_counts <- dplyr::full_join(ds_counts, tot_counts, by = "doc_id")

ds_counts <- ds_counts %>%
  dplyr::mutate_if(is.numeric, list(~./tot_counts), na.rm = TRUE) %>%
  dplyr::mutate_if(is.numeric, list(~.*100), na.rm = TRUE)

ds_counts$tot_counts <- NULL
```

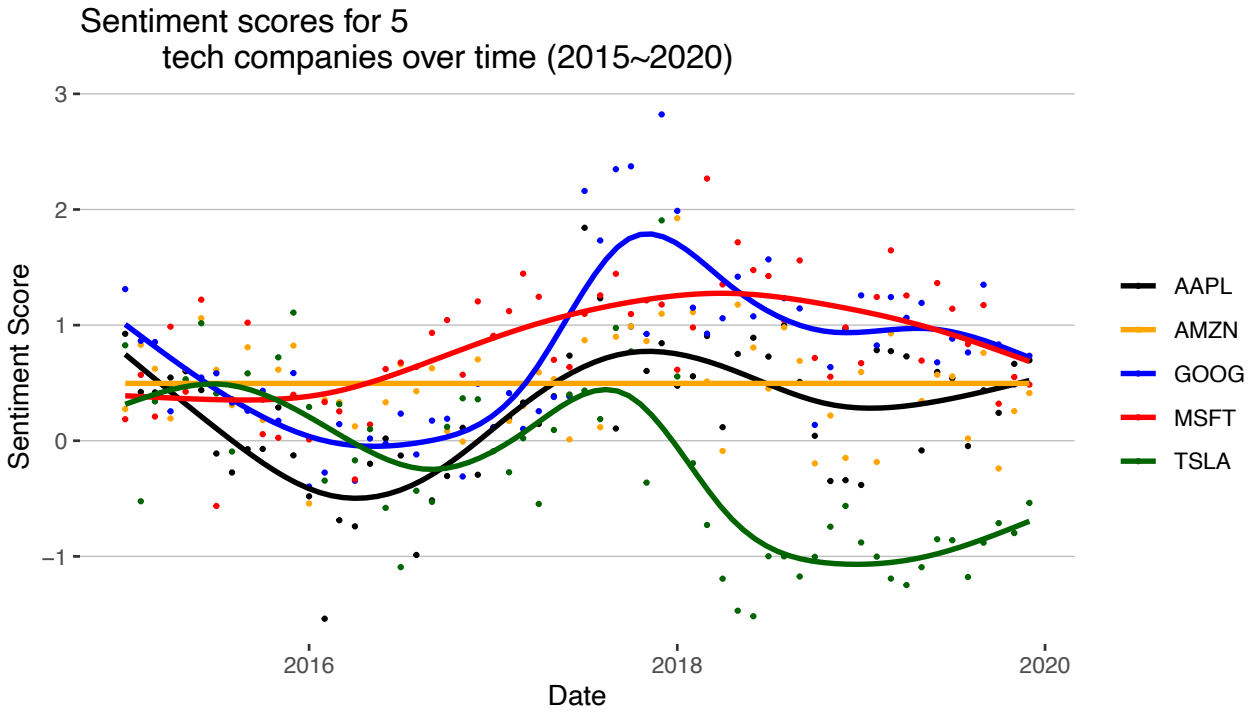
```r
# Simplify table to ticker_symbol, date and sentiment_score
twt_sentiment <- ds_counts %>%
  mutate(
    ticker_symbol = str_extract(doc_id, "^[A-Z]+"),
    date = as.Date(paste0(word(doc_id, 2, sep = "_"), '-01'), format='%Y-%m-%d')
    ) %>%
  dplyr::select(ticker_symbol, date, sentiment_score) %>%
  filter(date >= "2015-01-01")
```

Graphing time series plot

```r
# Graphing the time series plot
ggplot(twt_sentiment, aes(x=date, y=sentiment_score, color=ticker_symbol)) +
    geom_point(size = .5) +
    geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), size=1,
                level=0.95, se=T) +
    labs(x="Date", y = "Sentiment Score", title="Sentiment scores for 5
         tech companies over time (2015~2020)")+
    theme(panel.grid.minor.x=element_blank(),
          panel.grid.major.x=element_blank()) +
    theme(panel.grid.minor.y =   element_blank(),
          panel.grid.major.y =   element_line(colour = "gray",size=0.25)) +
    theme(rect = element_blank()) +
    theme(legend.title=element_blank()) +
    scale_color_manual(values = c("black",
                                  "orange",
                                  "blue",
                                  "red",
                                  "darkgreen"))
```




```
# Graphing the time series plot with confidence intervals
ggplot(twt_sentiment, aes(x=date, y=sentiment_score, color=ticker_symbol)) +
    geom_point(size = .5) +
    geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), size=1,
                level=0.95, se=F) +
    labs(x="Date", y = "Sentiment Score", title="Sentiment scores for 5
         tech companies over time (2015~2020)")+
    theme(panel.grid.minor.x=element_blank(),
          panel.grid.major.x=element_blank()) +
    theme(panel.grid.minor.y =   element_blank(),
          panel.grid.major.y =   element_line(colour = "gray",size=0.25)) +
    theme(rect = element_blank()) +
    theme(legend.title=element_blank()) +
    scale_color_manual(values = c("black",
                                  "orange",
                                  "blue",
                                  "red",
                                  "darkgreen"))
```


Sentiment scores for 5
tech companies over time (2015~2020)
Sentiment Score
3
2
1
0
−1
2016
2018
2020
Date
AAPL
AMZN
GOOG
MSFT
TSLA