Sentiment Time Series Analysis of Historical Tweets on Big-Tech Companies

Lee, Woo Chan
woochanl@andrew.cmu.edu

Department of Statistics and Data Science, Carnegie Mellon University

November 2021

Introduction

Fast growing "Big-Tech" companies tend to be extremely susceptible for fluctuations in public opinion. There could be countless reasons such as new releases in products, privacy concerns or scandals, and even new government regulations that may impact a person's view of the company. One common way that people express these shifts in opinions are through tweets. In this CBE, I wanted to explore the sentimental trends over time for selected Big-Tech companies and investigate whether changes in public sentiment can be accurately reflected in tweets.

Data

The dataset was part of a paper published in the 2020 IEEE International Conference under the Intelligent Data Mining track, primarily to determine possible speculators and influencers in the stock market. The dataset contains over 3 million unique tweets with features such as tweet id, post date, text body, and the number of comments and likes matched with the related company ticker. More specifically, the company tickers were restricted to 5 Big-Tech Companies: AAPL(Apple), AMZN(Amazon), GOOG(Google), MSFT(Microsoft), and TSLA(Tesla). In addition, the dates of the collected tweets spanned from 2014 December to 2019 December.

The token composition of the tweets corpora for each of the company tickers is shown in Table 1.

Company Ticker	Tokens
AAPL	10816314
AMZN	5627103
GOOG	5184923
MSFT	2886835
TSLA	10846623
Total	**35361798**

Table 1: Token composition per company ticker

Methods

The entire corpora of combined tweets was initially tokenized into single strings. During this process, I excluded stopwords, URLs, emojis, and additional typographical symbols such as hashtags ("#") and direction signs ("@"). Not only that, any common grammatical variation forms of the 5 company names were also excluded in order to give more focus to important information present in the corpora.

Document-frequency matrices were generated for each of the 5 companies in order to carry out descriptive analysis of the frequencies and keyness of the tokens. The log-likelihood (LL) measure was primarily used to identify semantic clusters or categories of features that showed statistically significant differences between the 5 company corpora.

On a separate note, the tokens were also grouped into each company tickers, by each month throughout 2014 December to 2019 December. This enabled the possibility of analyzing groups of tokens for each company over monthly periods of time.

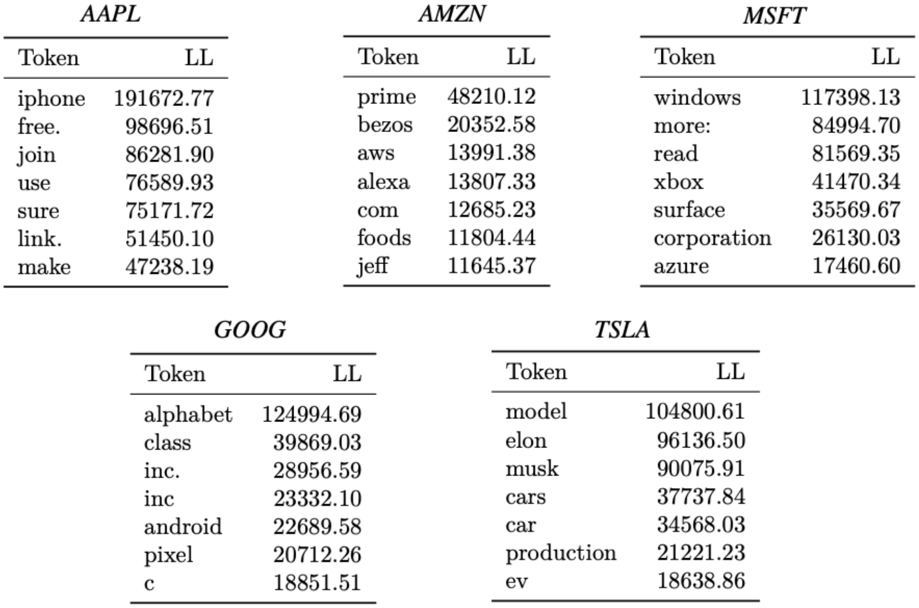
Next, the tokens were tagged using Docuscope, a dictionary-based rhetorical tagger. Since we were interested in sentiment analysis, the two categories in Docuscope that were primarily used were the positive and negative tags. These two categories were clusters that referenced the dimensions of positivity and negativity of a token respectively. The ”sentiment score” for a subset of tokens was defined to be the difference between the normalized frequencies of positive and negative-category tokens, as seen in the equation below. Eventually, each of the 5 company tickers had a sentiment score associated with each month of tweets.

$$Sentiment\ Score = Normalized\ count\ of\ positive\ category - Normalized\ count\ of\ negative\ category$$

In order to implement the time series plot, the sentiment scores for each company tickers were graphed through time. The peaks and troughs technique was used (Gabrielatos & Marchi 2012) to investigate the uptrend and downtrend of the sentiment scores throughout the graph.

Results

Table 2 shows the top 7 keywords for each of the 5 corpora containing tweets related to the company tickers. Each table uses a specific company ticker as the target while setting the rest of the corpora as the reference. From the tables it is evident that the keywords with high log-likelihood values are heavily related to the specific products and services associated with the company. The top keyword for AAPL, AMZN, MSFT, GOOG and TSLA turned out to be ”iphone”, ”prime”, ”windows”, ”alphabet”, and ”model” respectively.



AAPL

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

AMZN

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

MSFT

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

GOOG

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

TSLA

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

Table 2: Keyness table for the 5 company tickers

Figure 1 shows the time series plot for sentiment scores associated with the 5 big- tech companies. Figure 2 shows an identical plot with 95% confidence intervals (CI) applied. Both graphs show the sentiment scores in consecutive months during the period of 2014 December to 2019 December. The non-linear regression lines helped to identify a major trough for most companies close to the year of 2016, and a significant uptrend between 2017 and 2018. This suggests that there might have been certain global issues around 2016 and 2018 that caused public sentiments of the tech companies to undergo a major shift.

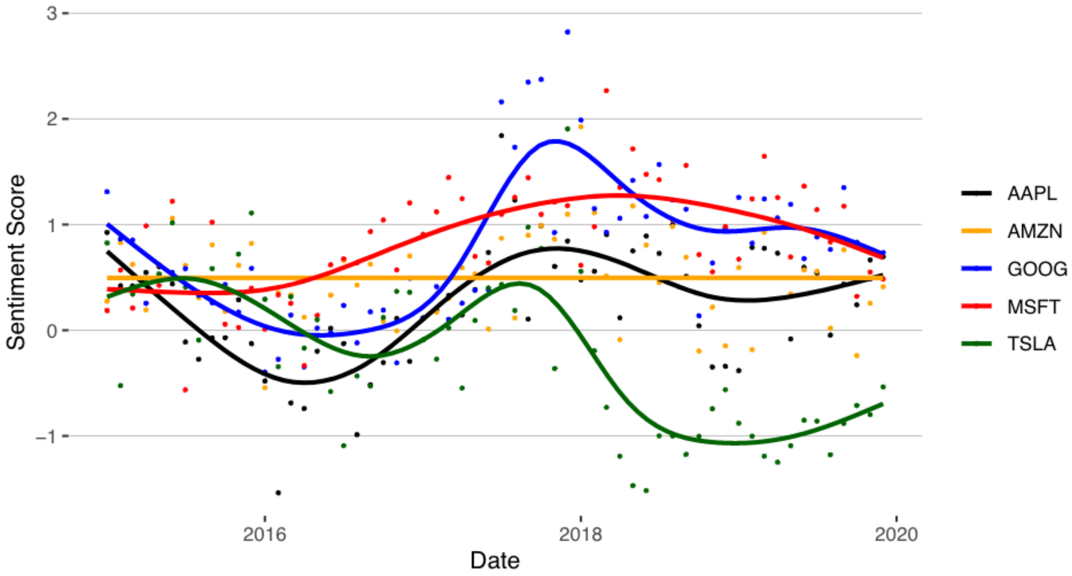


Figure 1: Sentiment scores over time

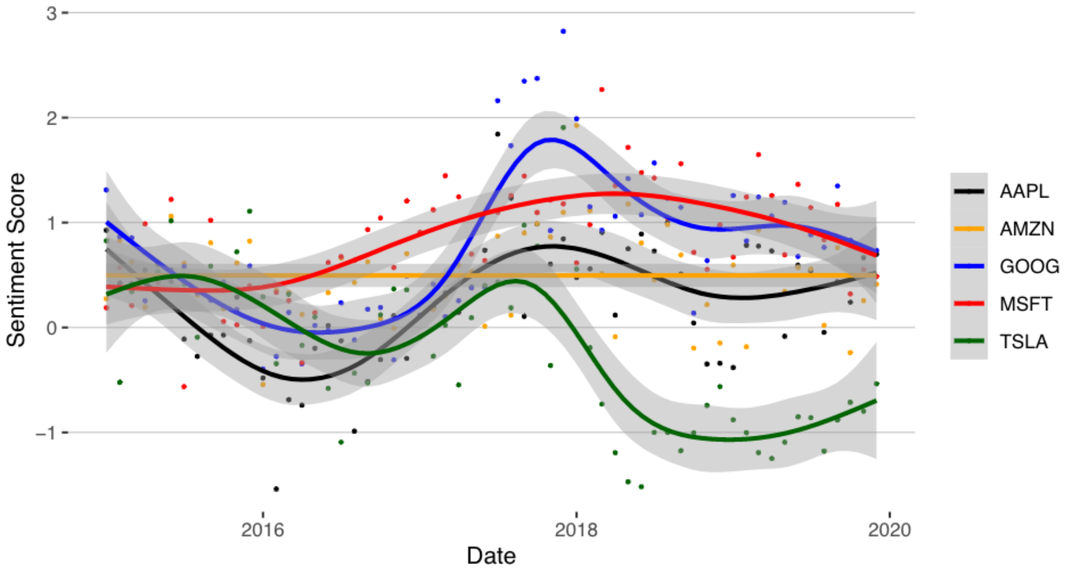


Figure 2: Sentiment scores over time with 95% CI

The year 2016 was the presidential election year for the US, and political uncertainty might have led to overall increased market volatility [1]. 2016 was also the year when there was a large stock market sell-off, one of the main reasons being the large drop in oil prices, as well as bad headlines from China concerning its slower growth prospects [2]. The overall market had a negative outlook during this time, and it is understandable that the public opinion and outlook on big-tech companies were also heavily impacted.

The year 2017 turned out to be a booming stock market as a result of resurgent economic growth and big corporate profits. The main reason behind this was likely the sweeping corporate tax cuts President Trump signed into the law [3]. With a booming economy and soaring consumer confidence, many of the big-tech companies saw huge increases in sales growth towards 2018. This seemingly translated to the sentiment scores of tweets, with the polarity curve for AAPL, GOOG, MSFT and TSLA showing an overall increasing trend during this period of time.

After 2018, the sentiment scores showed a decreasing trend for the majority of the companies, many of them reaching another trough towards 2019. TSLA showed the largest drop in sentiment scores, with the general public opinion suggesting a negative outlook. The sentiment score for AMZN however, remained relatively constant throughout time.

Discussion

Many of the Big-Tech companies surprisingly showed similar uptrend and downtrend in sentiment scores throughout the 5 years span. The specific polarity patterns were interpretable in terms of the overall market trends and global issues taking place in that particular point in time. With interesting slope patterns aligning with public opinions rather accurately, it was reasonable to state that changes in public sentiment was reflected in tweets fairly well.

In the future, it would be interesting to explore the linguistic features or categories that caused dynamic movements and huge drops in sentiment score for the Tesla tweets corpus, and how these compare with that of Microsoft, which showed a relatively steady increase towards a positive polarity over time. A good direction to take in the future would be to focus on the two corpora and explore clustering methodologies or multi-dimensional analysis to investigate linguistic features that co-occur in each corpus and how these relate to the sentimental pattern of the corpora over time.

References

[1] Jay Jenkins . (2016) . ”3 Reasons the Market Has Crashed in 2016” . The Motley Fool .
https://www.fool.com/investing/general/2016/01/29/3-reasons-the-market-has-crashed- in-2016-and-why-t.aspx

[2] Andy MacMillan . (2016) . ”What happened to the Great Tech Crash of 2016?” . Venturebeat
https://venturebeat.com/2016/09/18/what-happened-to-the-great-tech-crash-of-2016/

[3] Matt Egan and Danielle Wiener-Bronner (2017). ”It was an epic year for stocks” . CNN .
https://money.cnn.com/2017/12/29/investing/stocks-2017-wall-street/index.html

Code Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
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)
# Read main and meta data
twt <- read_csv("/Users/lee14257/Development/CMU/Text Analysis/Project/archive/Tweet.csv")
#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.c
meta <- read_csv("/Users/lee14257/Development/CMU/Text Analysis/Project/archive/Company_Tweet.csv")

# 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)

# 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)

# Create token
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"))

# docvar
doc_ticker <- names(twt_tkn) %>%
  data.frame(ticker = .)

docvars(twt_tkn) = doc_ticker

# Create dfm
twt_dfm <- twt_tkn %>% dfm()

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

kableExtra::kbl(twt_comp, caption = "Composition of the twitter corpus",
                booktabs = T, linesep = "") %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic() %>%
  kableExtra::row_spec(5, hline_after = TRUE) %>%
  kableExtra::row_spec(6, bold=T)

# Create keyness tables
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)

kableExtra::kbl(head(aapl_kw, 7), caption = "Tokens with the highest keyness
                values in the AAPL ticker compared to the rest", booktabs = T,
                linesep = "", digits = 2) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()

kableExtra::kbl(head(amzn_kw, 7), caption = "Tokens with the highest keyness
                values in the AMZN ticker compared to the rest", booktabs = T,
                linesep = "", digits = 2) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()

kableExtra::kbl(head(msft_kw, 7), caption = "Tokens with the highest keyness
                values in the MSFT ticker compared to the rest", booktabs = T,
                linesep = "", digits = 2) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()

kableExtra::kbl(head(goog_kw, 7), caption = "Tokens with the highest keyness
                values in the GOOG ticker compared to the rest", booktabs = T,
                linesep = "", digits = 2) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()

kableExtra::kbl(head(tsla_kw, 7), caption = "Tokens with the highest keyness
                values in the TSLA ticker compared to the rest", booktabs = T,
                linesep = "", digits = 2) %>%
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
# Add docuscope dictionary
ds_dict <- dictionary(file = "/Users/lee14257/Development/CMU/Text Analysis/Project/ds_dict.yml")

# Preprocessing twitter token table for time series
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')

# Tag 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

# 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 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"))
```