Text Analytics and Natural Language Processing (NLP)

A3: Business Insight Report

Hult International School

MsBA 4

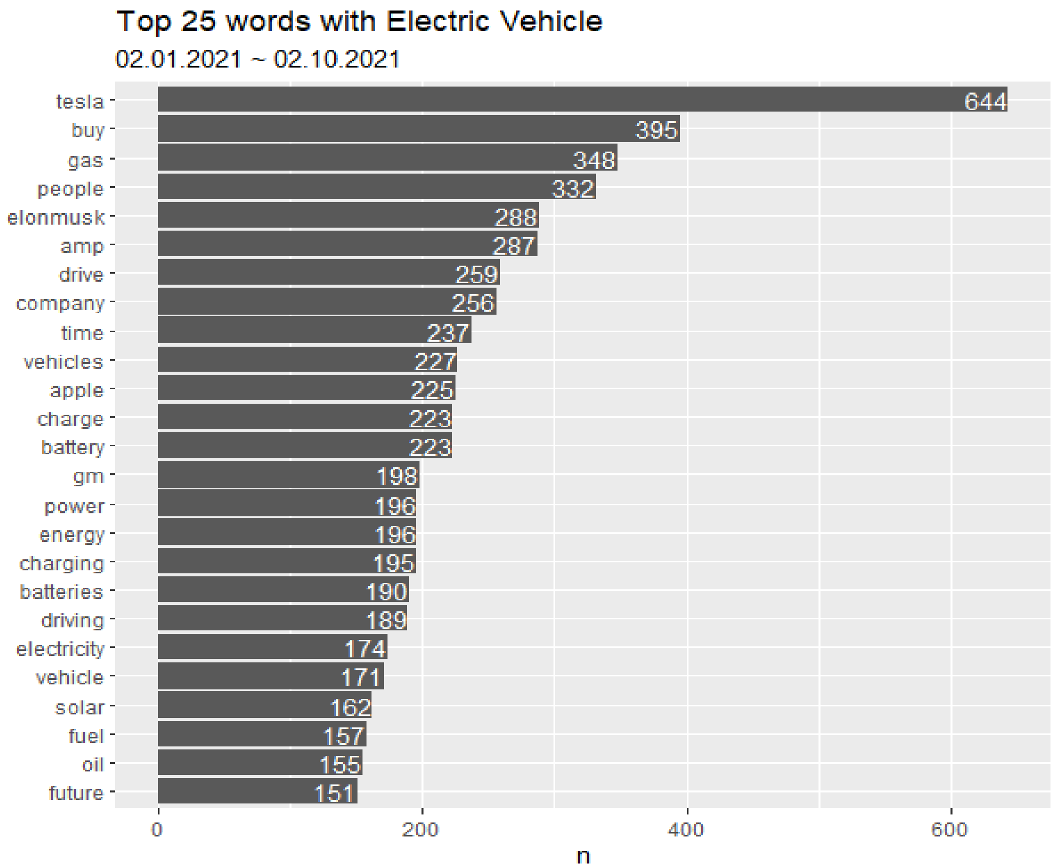
Jaeah Choi

Introduction

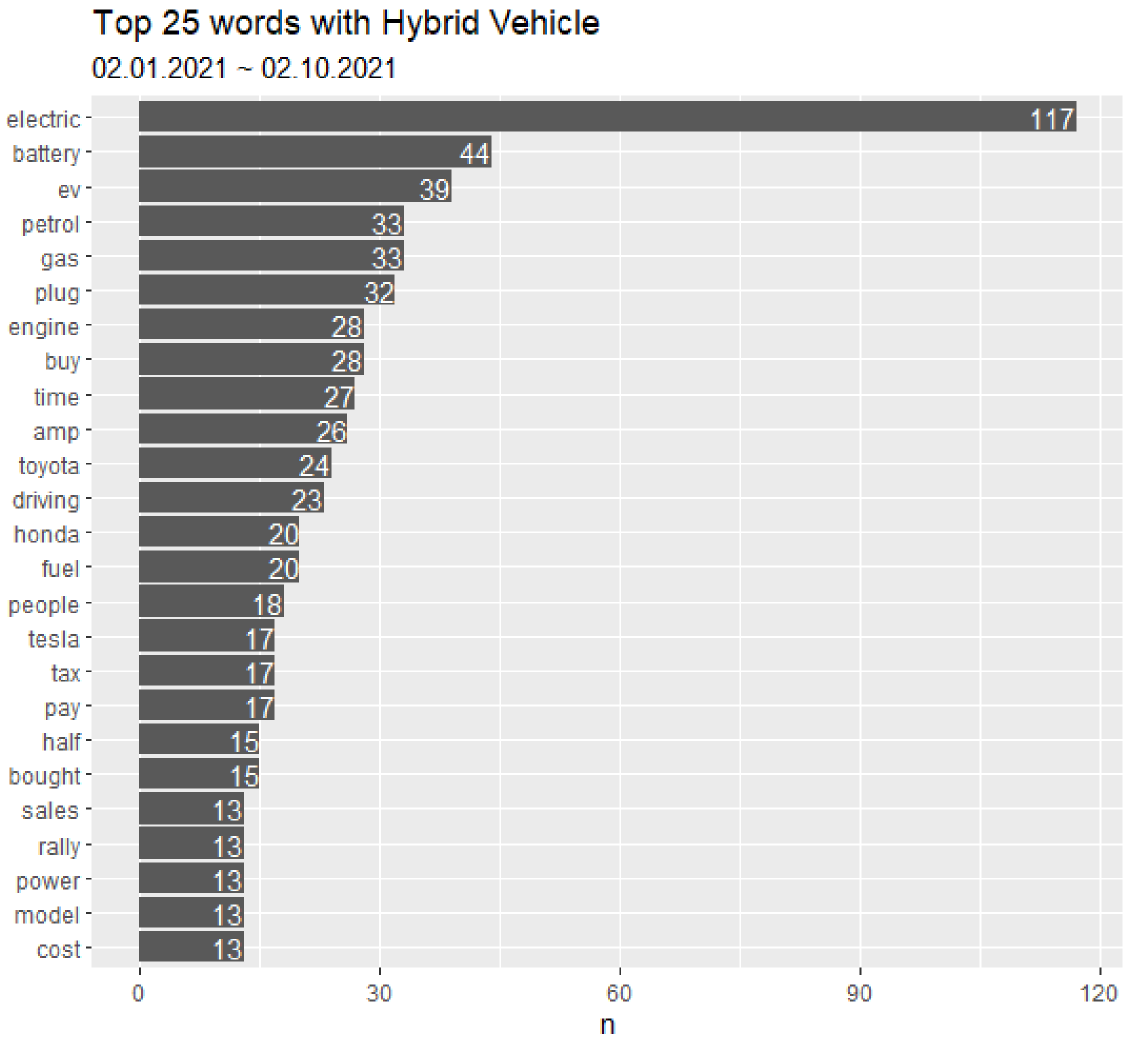
New technologies currently attracting attention in the automotive market are self-driving cars, flying cars, and electric vehicles. Along with technological advances, environmental issues have led many automakers to jump into developing electric vehicles. Besides, Apple[1] is discussing collaboration with automakers to launch new cars, and LG closed its mobile business and entered the electric car market. Governments in each country are also proposing new laws regarding electric vehicles. The purpose of this report is to understand people's perception of electric vehicles by comparing them with gasoline and hybrid vehicles with the aim of launching new electric cars. Moreover, it is intended to provide an analysis of Twitter users for securing potential customers by region and suggest business insights.

Analyzing Result

I have collected data from Twitter users to observe people's thinking about electric and hybrid vehicles. By analyzing the top 25 words with keywords, important companies in people's thoughts are able to identify and the new trend in the market can be measured. The charts below show the top 25 words people wrote on Twitter with electric vehicles and hybrid vehicles.



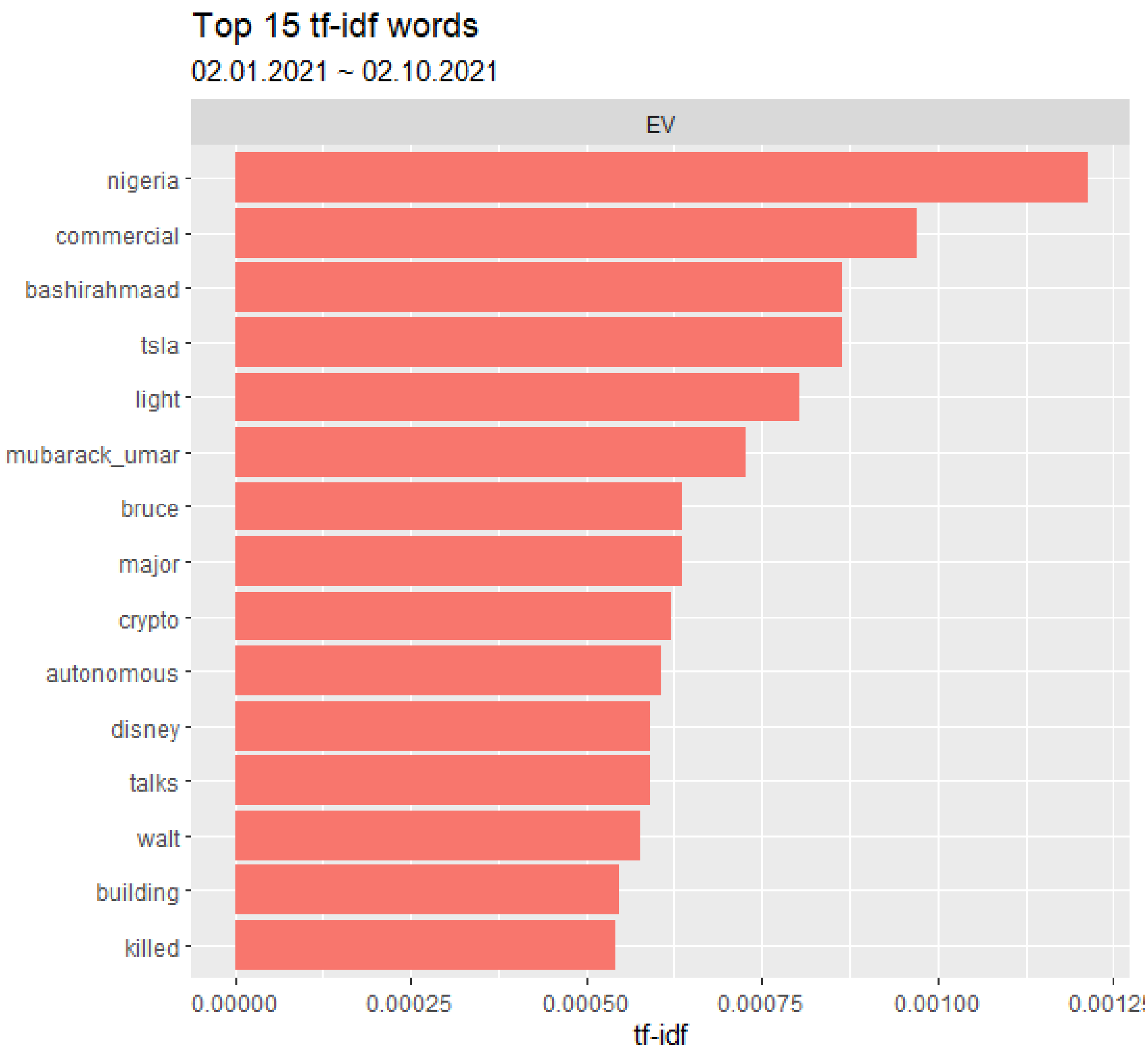
[Chart 1]



[Chart 2]

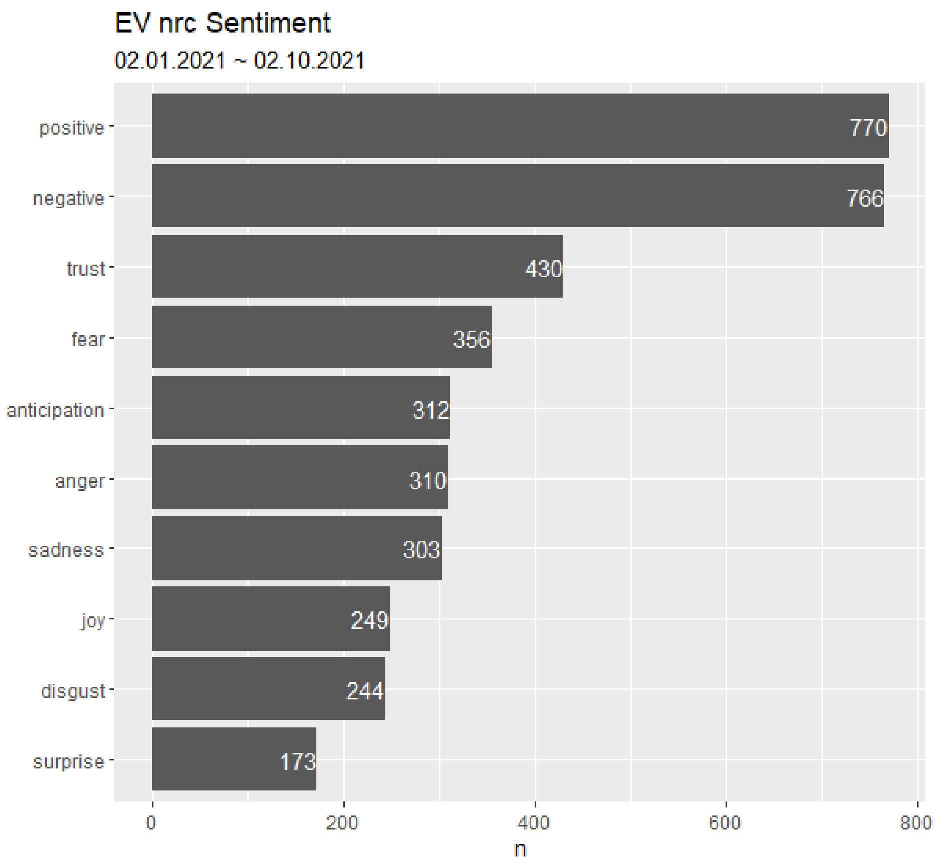
According to the chart1 above, it can be classified into three categories, which are big companies, fuel, and components. The companies highlighted with keywords are Tesla, Apple, and General Motors, which are leading to the market based on Twitter. After Tesla launched new electric cars in 2020, Teslar is the company that people think leading to the market and people are attention to Elon Musk's word who is CEO of Tesla. Also, General Motors got attention from people because of the advertising of Super Ball recently. Apple has been focused on worldwide potential customers after the company announced the new launch of new development electric cars. One of the electric car controversies is the battery issue, and it's shown in the chart. Likewise, words written with the hybrid vehicle have many overlapping words in char 1 such as Tesla, battery, and engine. The Japanese automobile companies, Toyota and Honda, are leading the hybrid cars. Also, the electric car's interests are shown in the chart2.

The charts below show the words that are of great importance in each category. By analyzing the top 15 TD-IDF words, the company can target potential customers by understanding their needs with appropriate marketing.



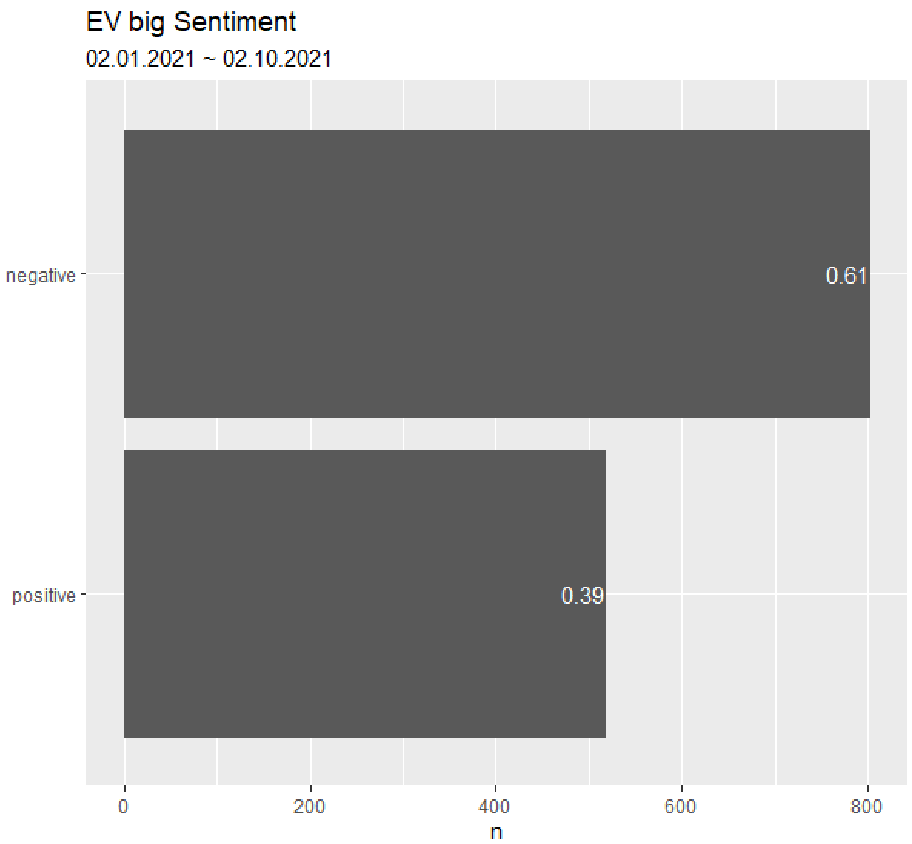
[Chart 3]

As shown in the graph above, Nigeria and Commercial are the top 2. Nigeria is one of the countries that potential growth is being talked about. By the importance of the word Nigeria shown as the top, the company can assume Nigerian people are interested in electric cars. Besides, many people are indirectly exposed to electric vehicles through commercials. The company should target Nigeria by gathering more data about why Nigerians are interested in an electric vehicle in a variety of ways so that the company provides available promotions. Also, advertising can be used as a marketing method and exposed to as many places as possible. Compare to the past, when the only convenience of movement was pursued, automobiles have become much smarter. Electric vehicles capable of autonomous driving will come out in the short future. The word Crypto in the chart shows that while incorporating AI into the car, the company must also pay attention to security.



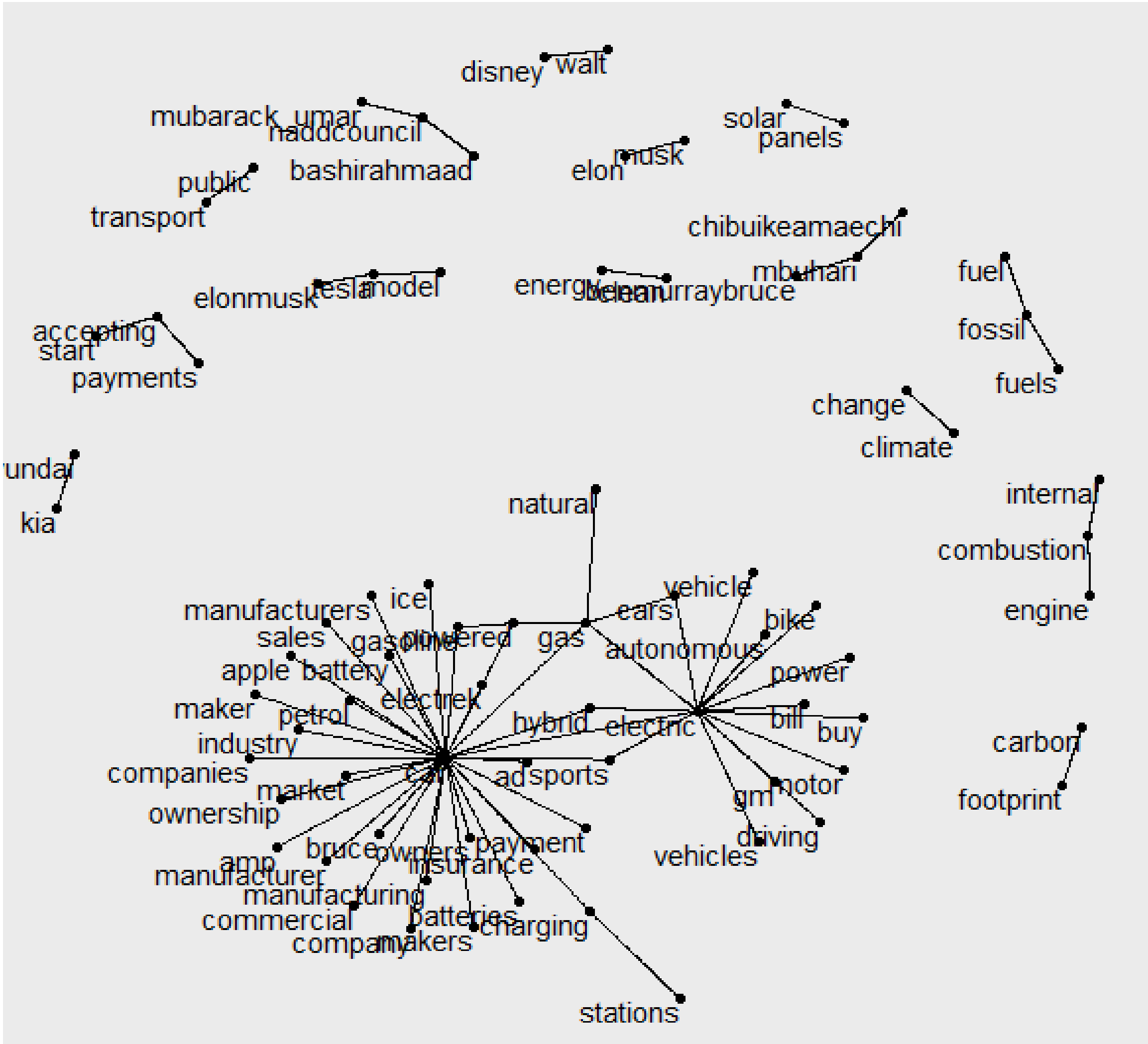
[Chart 4]

[Chart 5]



[Chart 6]

[Chart 7]



[Chart 8]

Chart 4 and 6 show frequent emotional words that people wrote with electric cars and Chart 8 shows the connection between words. According to charts 4 and 5 above, 60% of the words are negative written with electric cars. People anticipate the launch of new electric cars with new technologies from many automakers and becoming more common in our lives, but fear and negative views were seen more through the words. The company must analyze why people have a negative opinion about electric cars to understand future customer's concerns. For example, potential customers have a negative point of view because of battery and charging station issue. To be a leading automaker, developing a long-run battery and making the company's charging station or collaborate with a big gas station will be required. Additionally, using people's positive views of the electric vehicle will lead to maximizing commercial effect. There are 2 big nodes which are car and electric in Chart 6 above, and 2 nodes are connected with the words hybrid, sports, gas, power, and gasoline. Focusing on these connection words, the company can assume that power is one of the important conditions for buying a car. Also, the company can consider that customers tend to compare with gasoline and hybrid cars.

Conclusion

I analyzed potential customer's points of view about the electric vehicle using Twitter. As a result of the highest amount of people’s comments on Tesla, it can be seen that Tesla is leading the electric vehicle market in people’s perception. In addition, CEO Elon Musk’s high volume of comments indicates that his announcement and behaviors are receiving people’s attention. Because of the advertisement of General Motors’s electric cars during the half time of Super ball game, GM is mentioned a lot, which is an example showing the advertisement effect of the sports game that many people watch. Comparing with analyzing hybrid car results, battery and power are the customer’s main considerations when purchasing a general car. People have a negative view of electric cars in general. While expecting and trusting new technology cars, the words of fear and anger are written together. By analyzing TF-IDF, it was confirmed that Nigeria and Commercial have a large portion. If properly advertised in Nigeria through popular sports events and cultures such as Superball in the USA, it could attract potential customers. Moreover, it is necessary to prepare marketing that raises people’s interests in the product through witty advertisements. It’s also suggested to create a slogan with positive words about the electric vehicle that people think of.

Reference

1. Debby.W, River.D, Gabrielle.C and Kyunghee.P(2021). Who will build the Apple car?Here are candidate to watch. https://www.bloomberg.com/news/articles/2021-02-10/who-will-build-the-apple-car-here-are-candidates-to-watch

R Code

```
#setting library code
library(rtweet)
library(dplyr)
library(tidyverse)
library(tidytext)
library(stringr)
data(stop_words)
library(tidyr)

######################
## Electric Vehicle ##
######################
# collecting data from Twitter
EV_data <- search_tweets(
  "electric car", n=18000, include_rts = FALSE, lang ="en"
)
# filtering advertise & duplicated tweet
EV_clean <- EV_data %>%
              #subsetting showing data
            select('screen_name','text','source','favorite_count',
                   'retweet_count','hashtags') %>%
               # (assume having "http" as advertisement)
            filter(!str_detect(text, "https")) %>%
              # eliminate duplicated tweet
            group_by(screen_name) %>%
            distinct(text, .keep_all =T) %>%
            ungroup()
```

Deleting Numbers

```
library(tm)
EV_clean$text <- removeNumbers(EV_clean$text)

unnest_reg <- "([^A-Za-z_\\d#']|'(?![A-Za-z_\\d#]))"
```

EV tokenization

```
EV_token <- EV_clean %>%
        unnest_tokens(word, text,
                token = "regex", pattern = unnest_reg)%>%
        anti_join(stop_words)%>% #dropping stop words
        count(word, sort = T)
```

```
> EV_token
# A tibble: 15,140 x 2
   word         n
   <chr>    <int>
 1 car       5711
 2 electric  5500
 3 cars       861
 4 tesla      649
 5 buy        407
 6 gas        357
 7 people     338
 8 elonmusk   301
 9 amp        297
10 company    268
# ... with 15,130 more rows
```

#Token cleaning (word with no meaning or duplicated keywords)

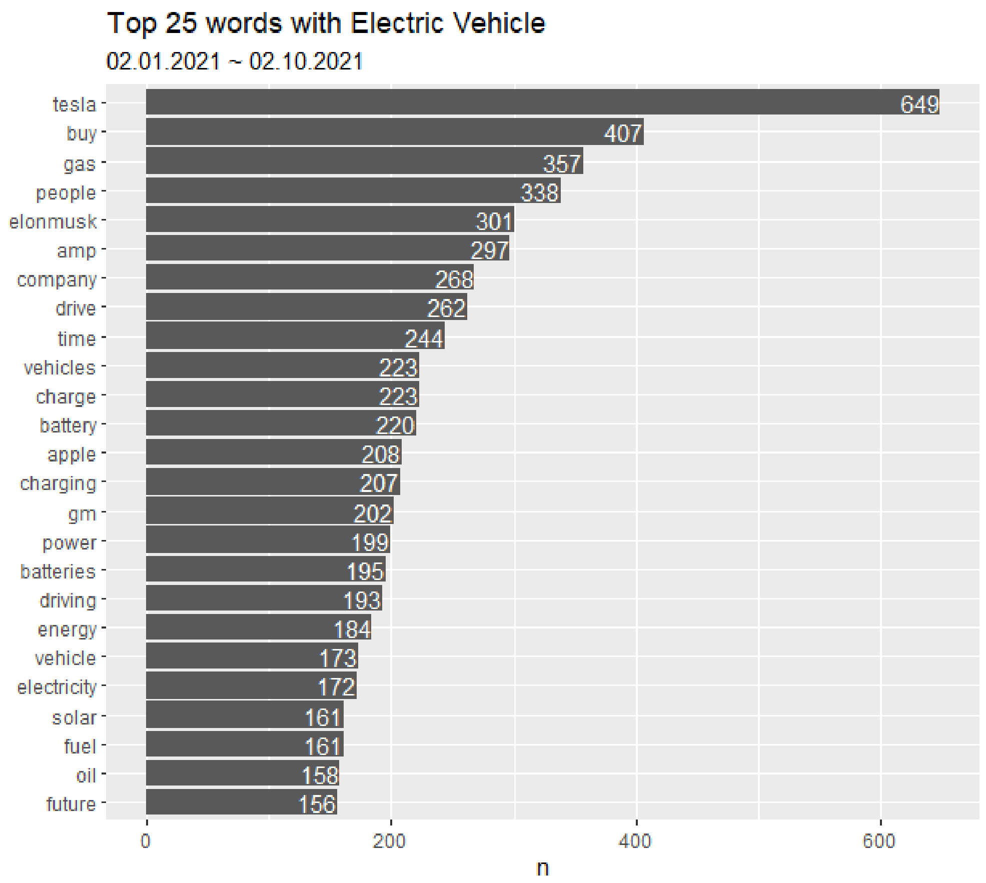
```
EV_token <- EV_token %>%
        filter(!str_detect(word, "don")) %>%
        filter(!str_detect(word, "ev"))
```

Add word proportion

```
EV_token_clean <- EV_token %>%
  mutate(word, n, proportion = (n/sum(n))*100)
```

```
# A tibble: 14,828 x 3
   word         n proportion
   <chr>    <int>      <dbl>
 1 car       5711      8.00
 2 electric  5500      7.70
 3 cars       861      1.21
 4 tesla      649      0.909
 5 buy        407      0.570
 6 gas        357      0.500
 7 people     338      0.473
 8 elonmusk   301      0.422
 9 amp        297      0.416
10 company    268      0.375
# ... with 14,818 more rows
>
```

```
# plotting top 25 words

library(ggplot2)

hist_EV_token <- EV_token_clean %>%

              filter(n<800 ) %>%

              top_n(25) %>%

              mutate(word = reorder(word,n )) %>%

              ggplot(aes(word, n))+

              geom_col()+

              geom_text(aes(label = comma(n, accuracy = 1)),

                    hjust =1.03, col='white')+

              labs(title = "Top 25 words with Electric Vehicle",

                 subtitle = "02.01.2021 ~ 02.10.2021",

                 x = NULL)+

              coord_flip()


print(hist_EV_token)
```

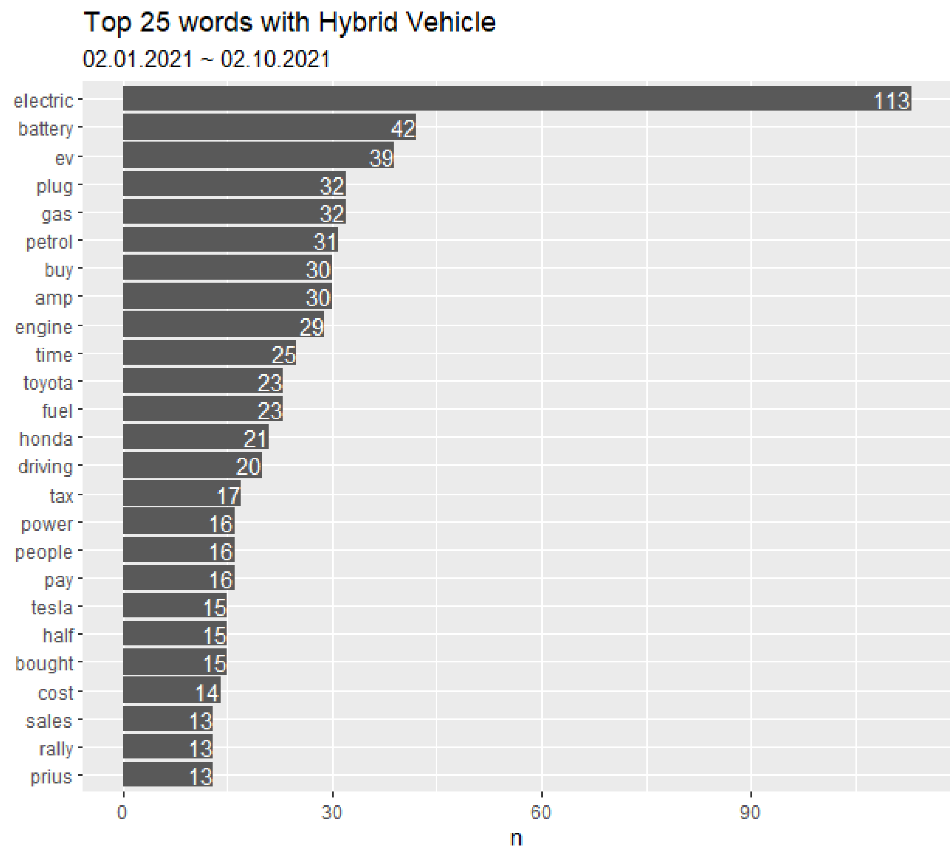



```
####################
## Hybrid Vehicle ##
####################
# collecting data from Twitter
hybrid_data <- search_tweets(
  "hybrid car", n=18000, include_rts = FALSE, lang="en"
)

# filtering advertise & duplicated tweet
hybrid_clean <- hybrid_data %>%

                  #subsetting showing data
                select('screen_name','text','source','favorite_count',
                       'retweet_count','hashtags') %>%
```

```
        # (assume having "http" as advertisement)
        filter(!str_detect(text, "https")) %>%

        # eliminate duplicated tweet
        group_by(screen_name) %>%
        distinct(text, .keep_all =T) %>%
        ungroup()

# dropping numbers in the text
hybrid_clean$text <- removeNumbers(hybrid_clean$text)

# Hybrid tokenization
hybrid_token <- hybrid_clean %>%
        unnest_tokens(word, text,
                token = "regex", pattern = unnest_reg)%>%
        anti_join(stop_words)%>%
        count(word, sort = T)

#Token cleaning (word with no meaning or duplicated keywords)
hybrid_token <- hybrid_token %>%
        filter(!str_detect(word, "car")) %>%
        filter(!str_detect(word, "ve"))
```

```
> hybrid_token
# A tibble: 2,992 x 2
   word         n
   <chr>    <int>
 1 car        522
 2 hybrid     475
 3 electric   113
 4 cars        47
 5 battery     42
 6 ev          39
 7 drive       38
 8 gas         32
 9 plug        32
10 petrol      31
# ... with 2,982 more rows
>
```

```
# plotting top 25 words

library(ggplot2)

hist_hybrid_token <- hybrid_token %>%
                    filter(n<400 ) %>%
                    top_n(25) %>%
                    mutate(word = reorder(word,n )) %>%
                    ggplot(aes(word, n))+
                    geom_col()+
                    geom_text(aes(label = comma(n, accuracy = 1)),
                              hjust =1.03, col='white')+
                    labs(title = "Top 25 words with Hybrid Vehicle",
                         subtitle = "02.01.2021 ~ 02.10.2021",
                         x = NULL)+
                   coord_flip()


print(hist_hybrid_token)
```




```
######################
## gasoline Vehicle ##
######################
# collecting data from Twitter
gasoline_data <- search_tweets(
  "gasoline car", n=18000, include_rts = FALSE, lang="en"
)

# filtering advertise & duplicated tweet
gas_clean <- gasoline_data %>%

          #subsetting showing data
         select('screen_name','text','source','favorite_count',
             'retweet_count','hashtags') %>%
```

```
  # (assume having "http" as advertisement)
  filter(!str_detect(text, "https")) %>%

  # eliminate duplicated tweet
  group_by(screen_name) %>%
  distinct(text, .keep_all =T) %>%
  ungroup()

gas_clean$text <- removeNumbers(gas_clean$text)

# Gasoline tokenization
gas_token <- gas_clean %>%
        unnest_tokens(word, text,
                token = "regex", pattern = unnest_reg)%>%
        anti_join(stop_words)%>%
        count(word, sort = T)
```

```
> gas_token
# A tibble: 2,352 x 2
   word         n
   <chr>    <int>
 1 car        365
 2 gasoline   333
 3 electric    46
 4 gas         42
 5 cars        36
 6 powered     35
 7 drive       26
 8 amp         24
 9 oil         24
10 ev          23
# ... with 2,342 more rows
```

```
########################
###### TD-IDF ###########
########################
# gathering 3 categories data
full_df <- bind_rows(EV_clean %>%
                mutate(text, category = 'EV'),
```

```
hybrid_clean%>%
  mutate(text, category = 'hybrid'),
gas_clean %>%
  mutate(text, category = 'gasoline'))
```

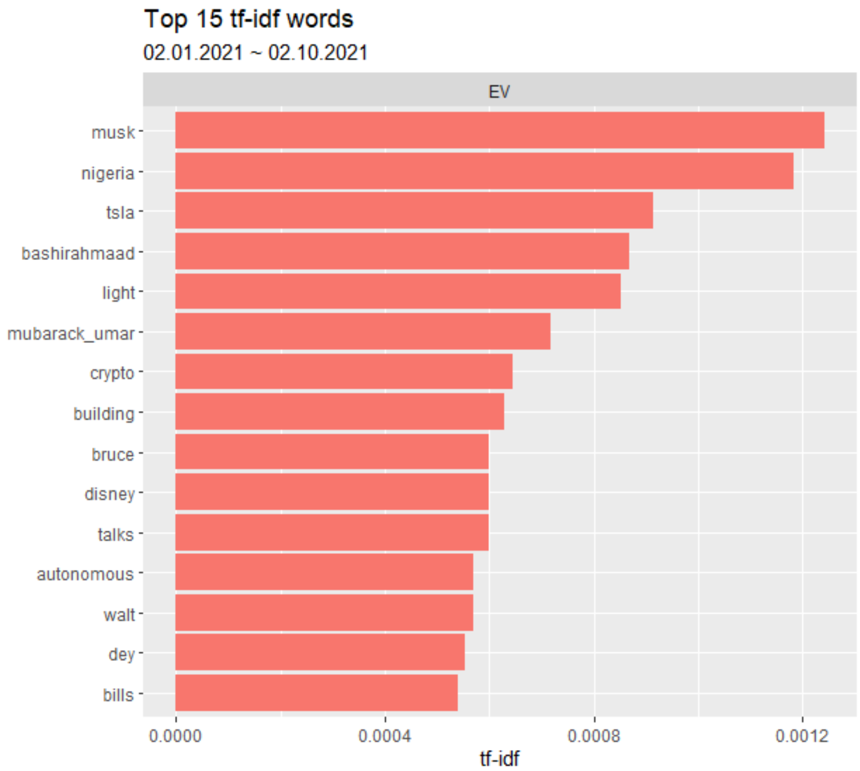
```
# tokenization
full_df_clean <- full_df %>%
        unnest_tokens(word, text) %>%
        anti_join(stop_words)%>%
        count(category, word, sort = T) %>%
        ungroup()
```

```
> full_df_clean
# A tibble: 20,542 x 3
   category word         n
   <chr>    <chr>    <int>
 1 EV       car       5710
 2 EV       electric  5513
 3 EV       cars       866
 4 EV       tesla      661
 5 hybrid   car        522
 6 hybrid   hybrid     481
 7 EV       buy        407
 8 gasoline car        364
 9 EV       gas        355
10 EV       people     336
# ... with 20,532 more rows
```

```
full_df_clean <- full_df_clean %>%
        bind_tf_idf(word, category, n)
```

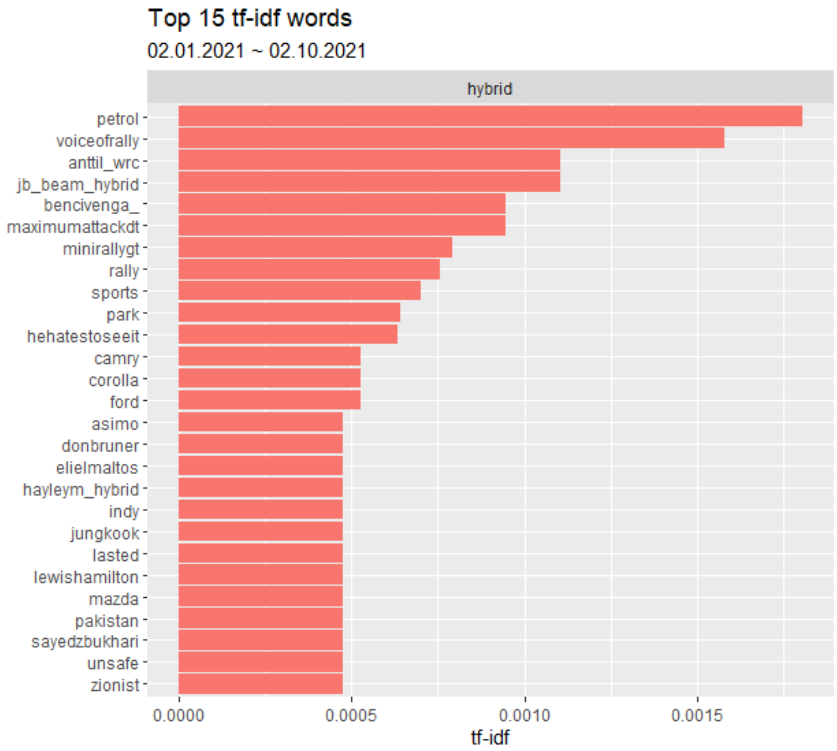
```
> full_df_clean
# A tibble: 20,542 x 6
   category word         n      tf   idf tf_idf
   <chr>    <chr>    <int>   <dbl> <dbl>  <dbl>
 1 EV       car       5710 0.0778      0      0
 2 EV       electric  5513 0.0751      0      0
 3 EV       cars       866 0.0118      0      0
 4 EV       tesla      661 0.00901     0      0
 5 hybrid   car        522 0.0751      0      0
 6 hybrid   hybrid     481 0.0692      0      0
 7 EV       buy        407 0.00555     0      0
 8 gasoline car        364 0.0745      0      0
 9 EV       gas        355 0.00484     0      0
10 EV       people     336 0.00458     0      0
# ... with 20,532 more rows
```

```
# tf-idf graphical approach EV
full_df_clean %>%
    arrange(desc(tf_idf)) %>%
    mutate(word=factor(word, levels =rev(unique(word)))) %>%
    group_by(category) %>%
    filter(category == 'EV') %>%
    filter(n<100) %>%
    top_n(15) %>%
    ungroup %>%
    ggplot(aes(word, tf_idf, fill=category))+
    geom_col(show.legend=FALSE)+
    labs(title = "Top 15 tf-idf words",
      subtitle = "02.01.2021 ~ 02.10.2021",
      x = NULL)+
    labs(x=NULL, y="tf-idf")+
    facet_wrap(~category, ncol=2, scales="free")+
    coord_flip()
```

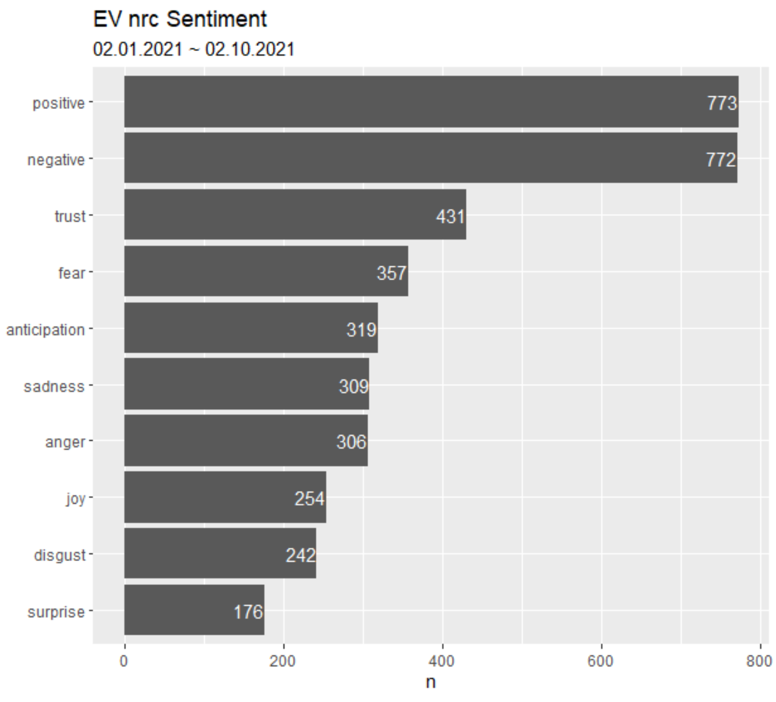


tf-idf graphical approach hybrid

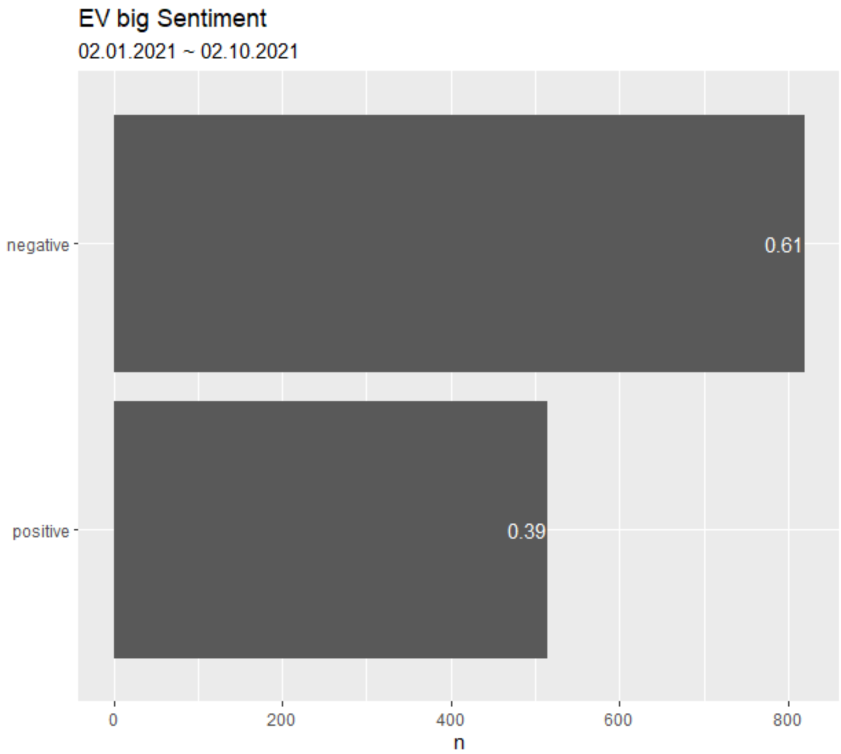
```
full_df_clean %>%
    arrange(desc(tf_idf)) %>%
    mutate(word=factor(word, levels =rev(unique(word)))) %>%
    group_by(category) %>%
    filter(category == 'hybrid') %>%
    filter(n<100) %>%
    top_n(15) %>%
    ungroup %>%
    ggplot(aes(word, tf_idf, fill=category))+
    geom_col(show.legend=FALSE)+
    labs(title = "Top 15 tf-idf words",
       subtitle = "02.01.2021 ~ 02.10.2021",
       x = NULL)+
    labs(x=NULL, y="tf-idf")+
    facet_wrap(~category, ncol=2, scales="free")+
    coord_flip()
```



```
################################
###### Sentiments ###########
################################
# NRC sentiment graph
full_df_clean %>%
  inner_join(get_sentiments("nrc")) %>%
  filter(category == "EV") %>%
  count(sentiment, sort=TRUE) %>%
  mutate(sentiment = reorder(sentiment,n )) %>%
  mutate(proportion = n/sum(n)) %>%
  ggplot(aes(sentiment, n))+
  geom_col()+
  geom_text(aes(label = comma(n, accuracy = 1)),
            hjust =1.03, col='white')+
  labs(title = "EV nrc Sentiment",
      subtitle = "02.01.2021 ~ 02.10.2021",
      x = NULL)+
  coord_flip()
```



```
# BING sentiment graph
full_df_clean %>%
  inner_join(get_sentiments("bing")) %>%
  filter(category == "EV") %>%
  count(sentiment, sort=TRUE) %>%
  mutate(percentage = n/sum(n)) %>%
  mutate(sentiment = reorder(sentiment,n )) %>%
  ggplot(aes(sentiment, n))+
  geom_col()+
  geom_text(aes(label = comma(percentage)),
            hjust =1.03, col='white')+
  labs(title = "EV big Sentiment",
       subtitle = "02.01.2021 ~ 02.10.2021",
       x = NULL)+
  coord_flip()
```

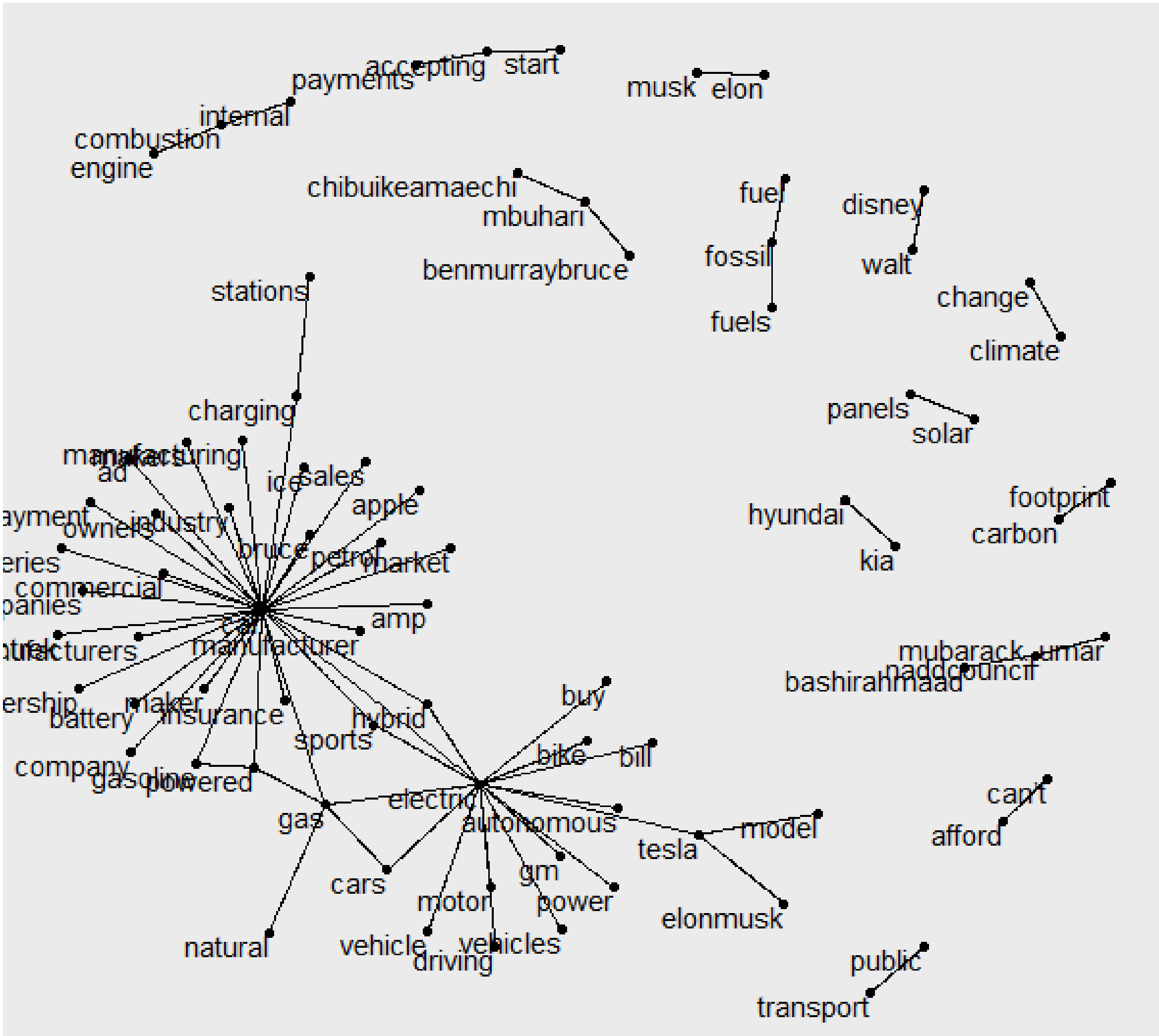


```
##########################
###### N-gram ###########
##########################
# bigram
car_bigram <- full_df %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2) %>%
  count(bigram, sort = TRUE) %>%
  separate(bigram, c("word1", "word2"), sep = " ")

# exclude stop words
bigrams_filtered <- car_bigram %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
```

```
> bigrams_filtered
# A tibble: 23,995 x 3
   word1    word2         n
   <chr>    <chr>     <int>
 1 electric car        3364
 2 electric cars        484
 3 electric vehicles    154
 4 car      company     134
 5 electric vehicle     109
 6 car      companies    88
 7 solar    panels       75
 8 hybrid   car          73
 9 car      batteries    70
10 elon     musk         69
# ... with 23,985 more rows
> 
```

```
# Graph
library(igraph)
library(ggraph)

#use lower n for less data
bigram_graph <- bigrams_filtered %>%
  filter(n>17) %>%
  graph_from_data_frame()
```

```
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```




```
##############################
###### word cloud ###########
##############################

library(wordcloud)
library(reshape2)
```

```
cloud_df <- full_df %>%
  group_by(category) %>%


  unnest_tokens(word, text)%>%
  filter(category == "EV") %>%
  anti_join(stop_words) %>%
  count(word, sort=T)
```

```
> cloud_df
# A tibble: 15,157 x 3
# Groups:   category [1]
   category word         n
   <chr>    <chr>    <int>
 1 EV       car       5710
 2 EV       electric  5513
 3 EV       cars       866
 4 EV       tesla      661
 5 EV       buy        407
 6 EV       gas        355
 7 EV       people     336
 8 EV       elonmusk   310
 9 EV       amp        297
10 EV       company    263
# ... with 15,147 more rows
>
```

```
cloud_df %>%
  inner_join(get_sentiments("nrc")) %>%
  mutate(percentage = n/sum(n)) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20","grey50"),
                   max.words=80, scale = c(1, 0.9))
```


```
#creating a sentiment word cloud for the bing library
cloud_df %>%
  inner_join(get_sentiments("bing")) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "gray80"),
                   max.words=100, scale = c(1, 0.9))
```