Text Mining for Sarcastic Comments from Reddit

2024-12-15

https://www.kaggle.com/datasets/sherinclaudia/sarcastic-comments-on-reddit?resource=download (https://www.kaggle.com/datasets/sherinclaudia/sarcastic-comments-on-reddit?resource=download)

```
library(readr)
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.4.2
```

```
library(widyr)
```

```
## Warning: package 'widyr' was built under R version 4.4.2
```

```
library(ggraph)
```

```
## Warning: package 'ggraph' was built under R version 4.4.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.4.2
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr   1.1.4     ✔ stringr 1.5.1
## ✔ forcats 1.0.0     ✔ tibble  3.2.1
## ✔ purrr   1.0.2     ✔ tidyr   1.3.1
```

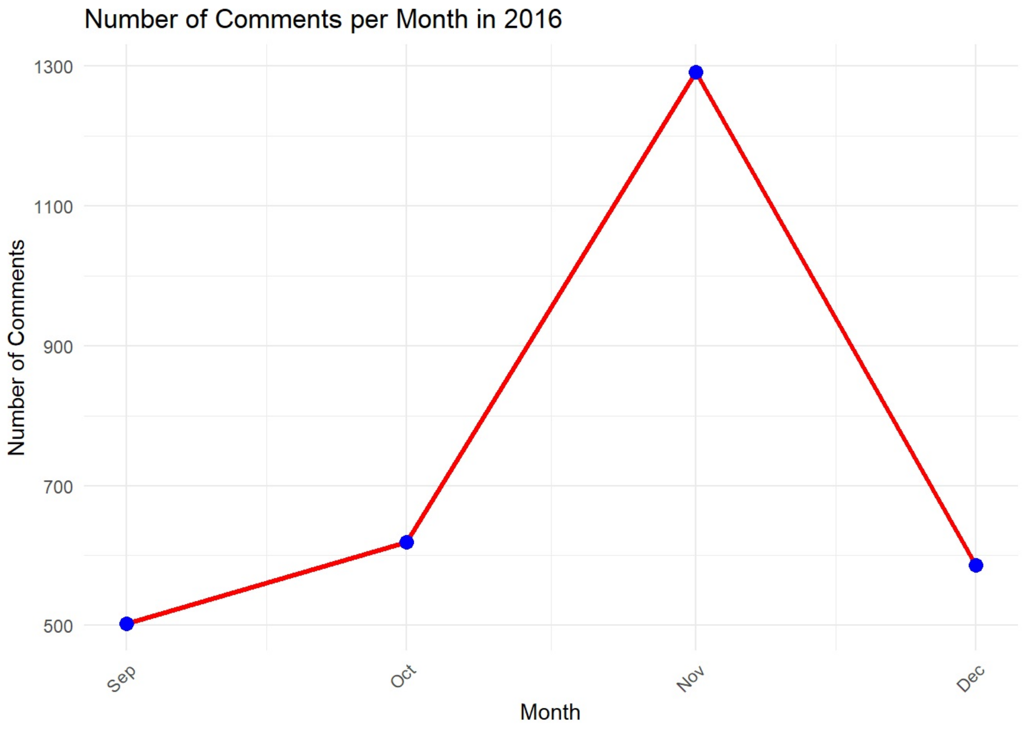
```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ igraph::%--%()         masks lubridate::%--%()
## ✖ dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## ✖ purrr::compose()       masks igraph::compose()
## ✖ tidyr::crossing()      masks igraph::crossing()
## ✖ dplyr::filter()        masks stats::filter()
## ✖ dplyr::lag()           masks stats::lag()
## ✖ purrr::simplify()      masks igraph::simplify()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data = read_csv("C:\\Users\\knc5576\\Downloads\\sarcasm.csv")
```

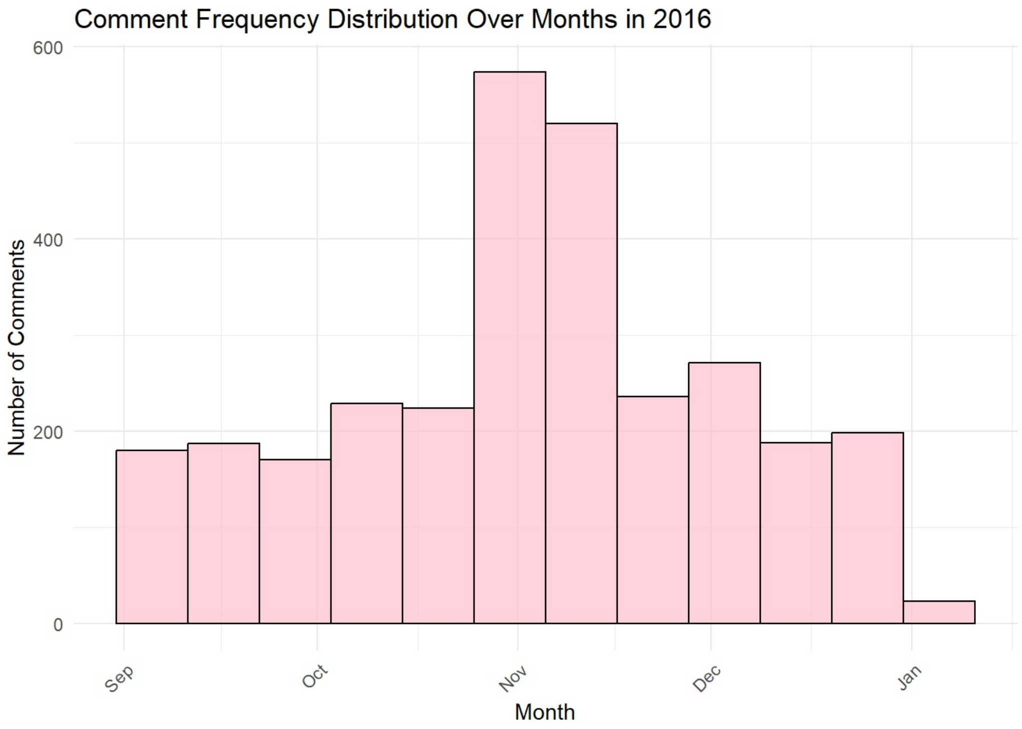
```
## New names:
## Rows: 3000 Columns: 11
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (5): comment, author, subreddit, date, parent_comment dbl (5): ...1, label,
## score, ups, downs dttm (1): created_utc
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...1`
```

```
data_2016 <- data %>%
  mutate(created_utc = as_datetime(created_utc)) %>%
  filter(year(created_utc) == 2016) %>%
  mutate(month = as.Date(floor_date(created_utc, "month")))
monthly_counts <- data_2016 %>%
  group_by(month) %>%
  summarise(comment_count = n())
ggplot(monthly_counts, aes(x = month, y = comment_count)) +
  geom_line(color = "red", size = 1.2) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Number of Comments per Month in 2016",
       x = "Month",
       y = "Number of Comments") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
ggplot(data_2016, aes(x = created_utc)) +
  geom_histogram(bins = 12, fill = "pink", color = "black", alpha = 0.7) +
  labs(title = "Comment Frequency Distribution Over Months in 2016",
       x = "Month",
       y = "Number of Comments") +
  scale_x_datetime(date_labels = "%b", date_breaks = "1 month") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
## Calculate frequency of comments
user_comment_freq <- data %>%
  group_by(author) %>%
  summarise(comment_count = n()) %>%
  arrange(desc(comment_count))
# top 10 users
head(user_comment_freq, 10)
```

```
## # A tibble: 10 × 2
##    author            comment_count
##    <chr>                     <int>
##  1 xVoltage360                   5
##  2 Disheartend                   4
##  3 JoanFoster                    4
##  4 Maryland_Mansion              4
##  5 ShyBiDude89                   4
##  6 Adam_Marx                     3
##  7 BabyJesusStig                 3
##  8 Brodoof                       3
##  9 CasualViewer24                3
## 10 Cthulhuonpcin144p             3
```

```
# Calculate the frequency of comments per subreddit
subreddit_comment_freq <- data %>%
  group_by(subreddit) %>%
  summarise(comment_count = n()) %>%
  arrange(desc(comment_count))
# View the top 10 subreddits with the most comments
head(subreddit_comment_freq, 10)
```

```
## # A tibble: 10 × 2
##    subreddit       comment_count
##    <chr>                   <int>
##  1 AskReddit                 255
##  2 politics                  211
##  3 The_Donald                117
##  4 nfl                        58
##  5 leagueoflegends            46
##  6 pcmasterrace               43
##  7 worldnews                  43
##  8 nba                        32
##  9 funny                      31
## 10 GlobalOffensive            29
```

```
data <- data %>%
  mutate(comment_id = row_number())
data_tokens <- data %>%
  unnest_tokens(word, comment) %>%
  anti_join(stop_words, by = "word")

# Create a pairwise count of word pairs within the same comment
word_pairs <- data_tokens %>%
  pairwise_count(word, comment_id, sort = TRUE)
head(word_pairs, 10)
```

```
## # A tibble: 10 × 3
##    item1  item2      n
##    <chr>  <chr>  <dbl>
##  1 lot    people     5
##  2 people lot        5
##  3 2      1          4
##  4 time   people     4
##  5 shit   people     4
##  6 white  people     4
##  7 sense  makes      4
##  8 makes  sense      4
##  9 people time       4
## 10 people shit       4
```

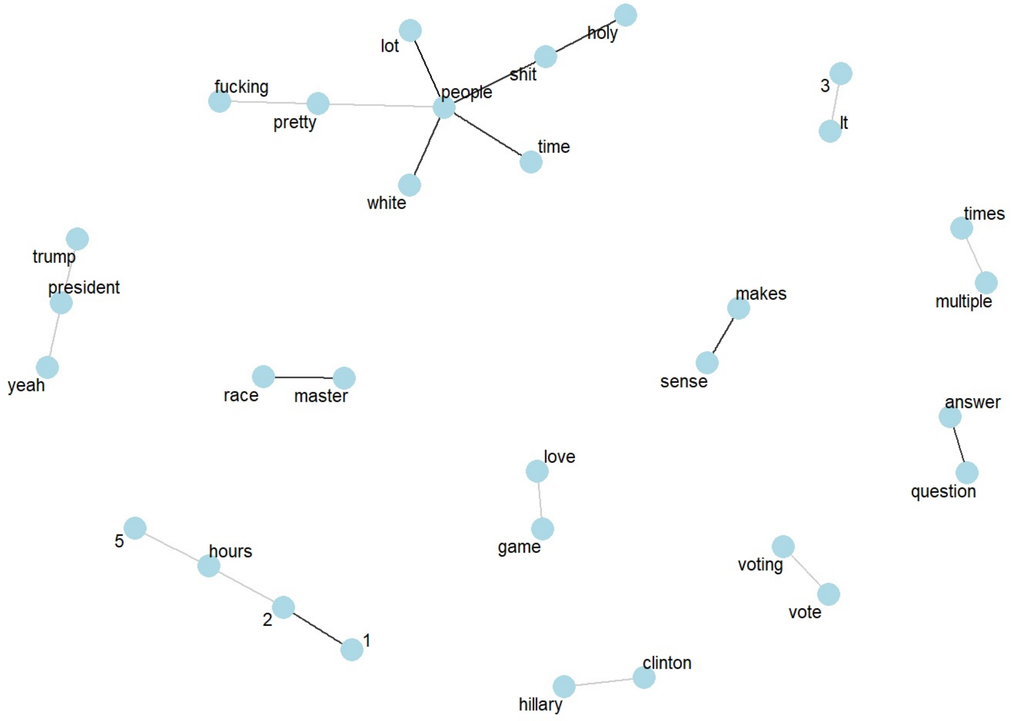
```
# Using ggraph, igraph, and ggplot2 to visualize the network of word pairs

head(word_pairs)
```

```
## # A tibble: 6 × 3
##   item1  item2      n
##   <chr>  <chr>  <dbl>
## 1 lot    people     5
## 2 people lot        5
## 3 2      1          4
## 4 time   people     4
## 5 shit   people     4
## 6 white  people     4
```

```
summary(word_pairs)
```

```
##     item1              item2                 n       
##  Length:58170       Length:58170       Min.   :1.00  
##  Class :character   Class :character   1st Qu.:1.00  
##  Mode  :character   Mode  :character   Median :1.00  
##                                        Mean   :1.01  
##                                        3rd Qu.:1.00  
##                                        Max.   :5.00  
```

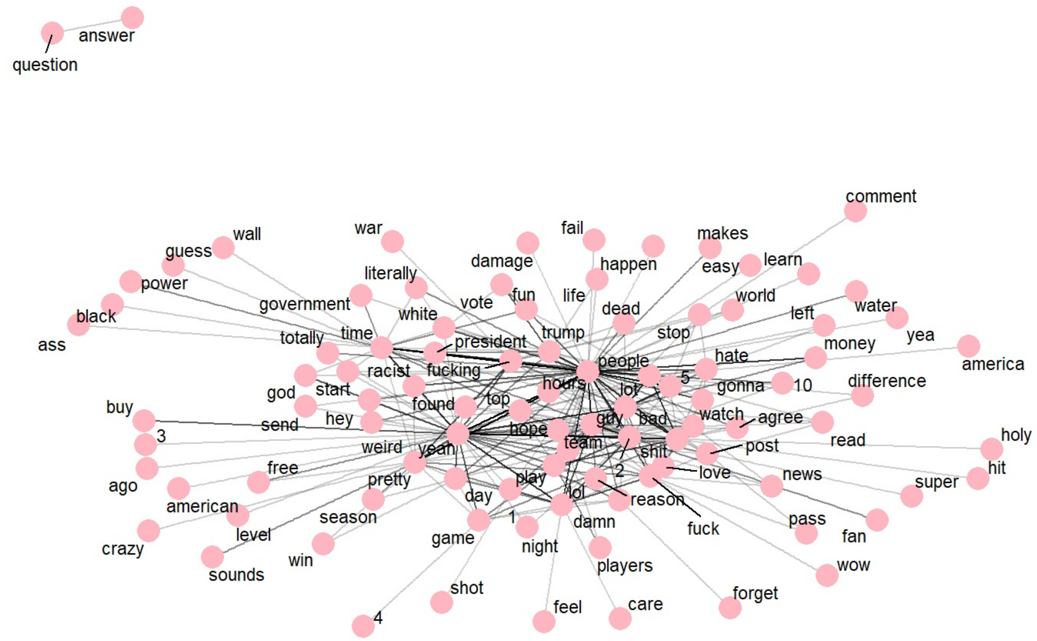
```
filtered_word_pairs <- word_pairs %>%
  filter(n > 2)
word_pairs_graph <- graph_from_data_frame(filtered_word_pairs)

ggraph(word_pairs_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), repel = TRUE, size = 3) +
  theme_void()
```



```
keyword_pairs <- data_tokens %>%
  pairwise_count(word, subreddit, sort = TRUE)
filtered_keyword_pairs <- keyword_pairs %>%
  filter(n > 4)

set.seed(1234)

filtered_keyword_pairs %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(alpha = n), show.legend = FALSE) +
  geom_node_point(color = "lightpink", size = 5) +
  geom_node_text(aes(label = name), repel = TRUE, size = 3, max.overlaps = 10) +
  theme_void() +
  labs(title = "Keyword Pair Network", subtitle = "Filtered for pairs with n > 2")
```

Keyword Pair Network

Filtered for pairs with n > 2



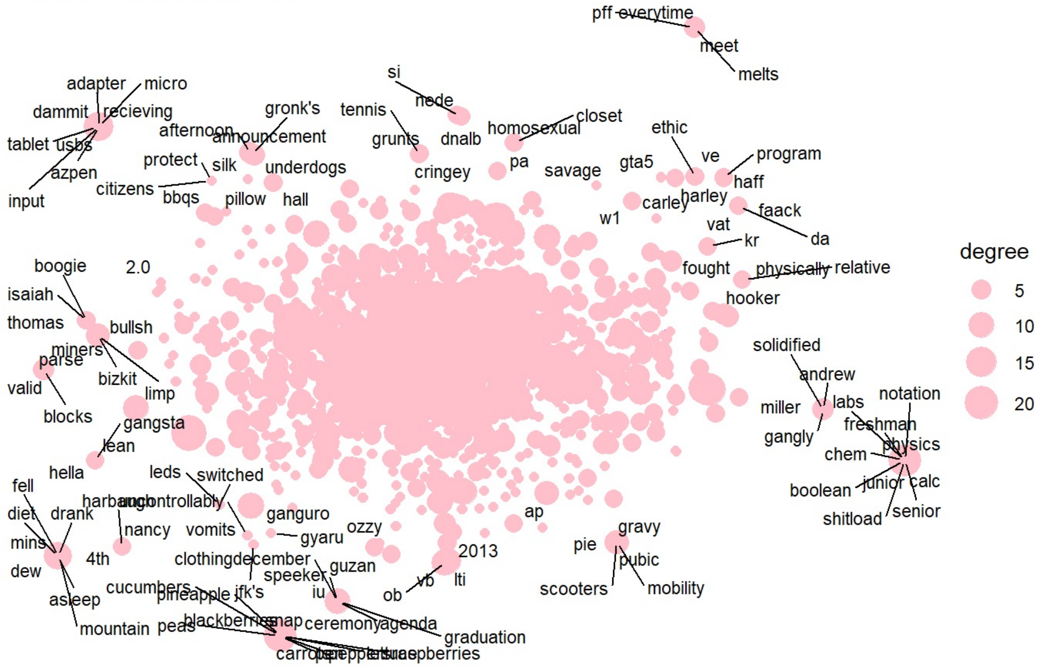
```
comment_cors <- data_tokens %>%
  group_by(comment_id) %>%
  filter(n() > 1) %>%
  pairwise_cor(word, comment_id, sort = TRUE)
```

```
set.seed(1234)

filtered_cors <- comment_cors %>%
  filter(correlation > 0.8)

graph <- graph_from_data_frame(filtered_cors)
V(graph)$degree <- degree(graph)

# Make plot
ggraph(graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation),
                 edge_colour = "purple", show.legend = FALSE) +
  geom_node_point(aes(size = degree), color = "pink") +
  geom_node_text(aes(label = name), repel = TRUE,
                 max.overlaps = 20, size = 3, point.padding = unit(0.2, "lines")) +
  scale_edge_width(range = c(0.2, 2)) +
  scale_edge_alpha(range = c(0.3, 0.9)) +
  scale_size(range = c(2, 8)) +
  theme_void() +
  labs(title = "Keyword Correlation Network",
       subtitle = "Filtered for Correlations > 0.8",
       edge_width = "Correlation",
       edge_alpha = "Correlation")
```

```
## Warning: ggrepel: 3203 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Keyword Correlation Network

Filtered for Correlations > 0.8


pff
everytime
meet
melts
si
nede
adapter
micro
dammit
recieving
tablet
usbs
azpen
input
gronk's
afternoon
announcement
tennis
grunts
dnalb
homosexual
closet
ethic
ve
program
haff
protect
silk
underdogs
citizens
bbqs
pillow
hall
cringey
pa
savage
gta5
carley
harley
faack
w1
vat
da
kr
fought
physically
relative
hooker
boogie
isaiah
thomas
bullsh
parse
miners
bizkit
limp
valid
blocks
gangsta
lean
solidified
andrew
miller
labs
notation
freshman
gangly
physics
chem
boolean
junior
calc
shitload
senior
hella
leds
switched
fell
harbaugh
uncontrollably
diet
drank
nancy
vomits
ganguro
mins
gyaru
ozzy
ap
4th
clothing
december
speeker
guzan
pie
gravy
dew
pubic
cucumbers
pineapple
jfk's
iu
ob
vb
lti
scooters
mobility
asleep
mountain
peas
blackberries
map
ceremony
agenda
graduation
carrots
peppers
lettuce
raspberries
degree