Sentiment Analysis

Christine Yuan

Setup

```r
library(readr)
library(tidytext)
library(textdata)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```r
library(ggplot2)
library(ggnewscale)
dataframe <- read_csv("twitch-chat-2236538812.csv", show_col_types = FALSE)
messages <- dataframe[, 4]
unlisted_messages <- unlist(messages)
vector_split_messages <- unlist(strsplit(unlisted_messages, split = " "))
split_messages <- data.frame(vector_split_messages)
names(split_messages) <- "Word"
```

NRC Custom Lexicon, Top 150 Most Used Words in Twitch Chat with Sentiment

```r
nrc_sentiment <- get_sentiments(lexicon = "nrc")
nrc_lexicon <- data.frame("Word" = nrc_sentiment$word, "Sentiment" = nrc_sentiment$sentiment)
nrc_lexicon <- nrc_lexicon[nrc_lexicon$Sentiment == "positive" | nrc_lexicon$Sentiment == "negative", ]
```

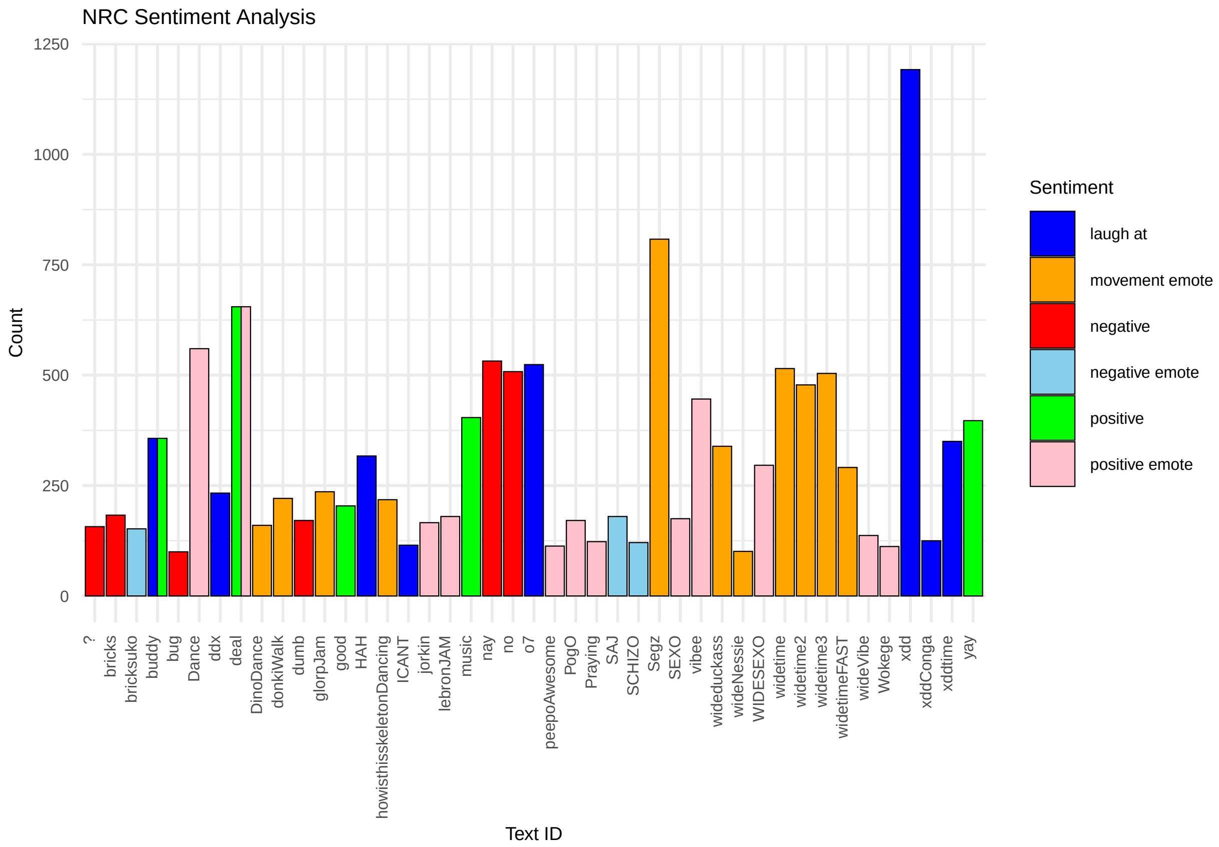
```r
nrc_twitch_words <- c("xdd", "Segz", "deal", "Dance", "o7", "widetime", "no", "widetime3", "widetime2",
nrc_twitch_words_sentiment <- c("laugh at", "movement emote", "positive emote", "positive emote", "laug
nrc_twitch_lexicon <- data.frame("Word" = nrc_twitch_words, "Sentiment" = nrc_twitch_words_sentiment)

nrc_new_lexicon <- rbind(nrc_lexicon, nrc_twitch_lexicon)
```

NRC Sentiment Analysis Graph

```r
# Join with sentiment lexicon to classify words
nrc_sentiment <- split_messages %>%
  inner_join(nrc_new_lexicon, by = "Word") %>%
  count(Word, Sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Warning in inner_join(., nrc_new_lexicon, by = "Word"): Detected an unexpected many-to-many relations
## i Row 20 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 3442 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```r
# Filter
filter_nrc_sentiment <- nrc_sentiment[nrc_sentiment$n >= 100, ]

# Plot the sentiment analysis results
filter_nrc_sentiment %>%
  ggplot(aes(x = factor(Word), y = n, fill = Sentiment)) +
  geom_col(position = "dodge", color = "black", linewidth = 0.2) +
  labs(title = "NRC Sentiment Analysis", x = "Text ID", y = "Count") +
  scale_fill_manual(values = c("positive" = "green", "negative" = "red", "movement emote" = "orange", "
  theme_minimal() +
  theme(text = element_text(size = 7),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0))
```


NRC Sentiment Analysis
Count
1250
1000
750
500
250
0
?
bricks
bricksuko
buddy
bug
Dance
ddx
deal
DinoDance
donkiWalk
dumb
glorpJam
good
HAH
howisthisskeletonDancing
ICANT
jorkin
lebronJAM
music
nay
no
o7
peepoAwesome
PogO
Praying
SAJ
SCHIZO
Segz
SEXO
vibee
wideduckass
wideNessie
WIDESEXO
widetime
widetime2
widetime3
widetimeFAST
wideVibe
Wokege
xdd
xddConga
xddtime
yay
Text ID
Sentiment
laugh at
movement emote
negative
negative emote
positive
positive emote

Flat Sentiment Analysis Graph (What happens if we don't create a custom lexicon?)

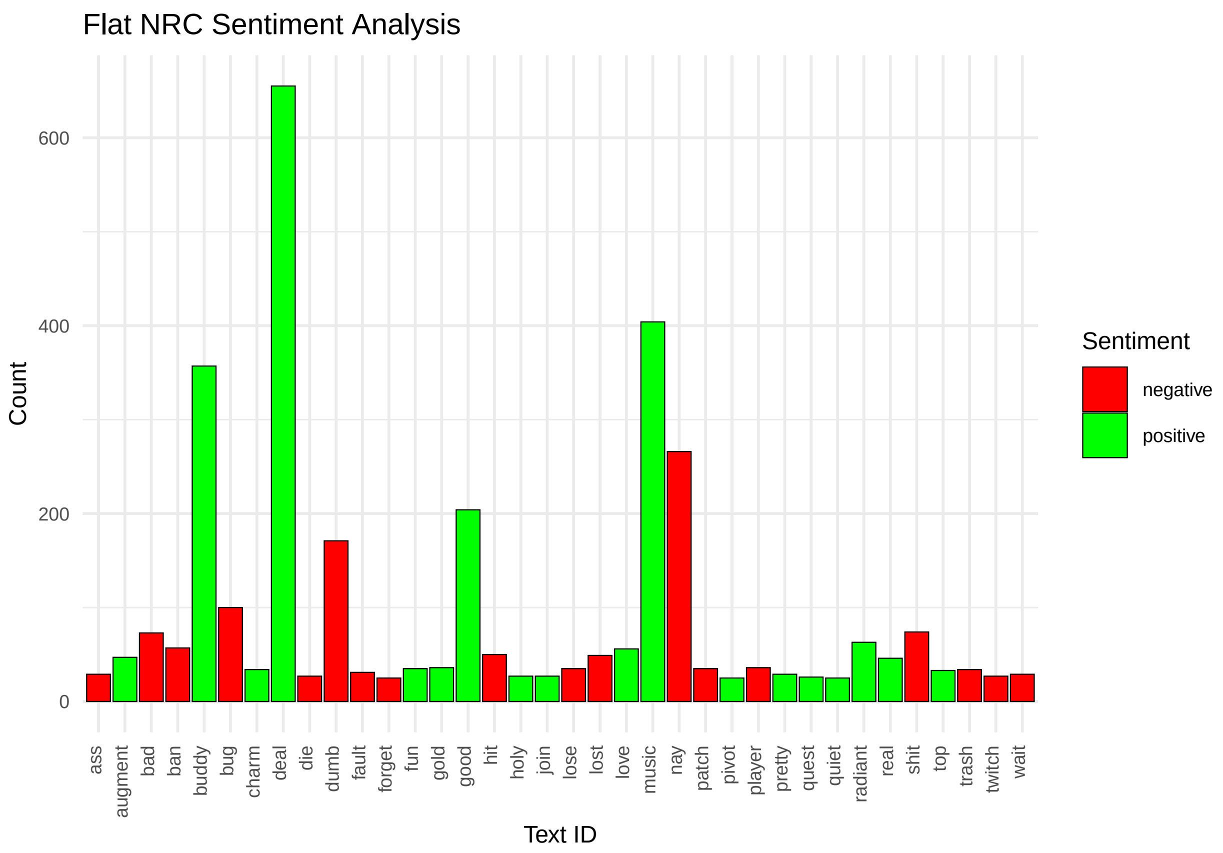
```
flat_nrc_messages <- semi_join(split_messages, nrc_lexicon)
```

```
## Joining with ‘by = join_by(Word)‘
```

```
flat_nrc_sentiment <- split_messages %>%
  inner_join(nrc_lexicon, by = "Word") %>%
  count(Word, Sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Warning in inner_join(., nrc_lexicon, by = "Word"): Detected an unexpected many-to-many relationship
## i Row 36 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 3442 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```
filter_flat_nrc_sentiment <- flat_nrc_sentiment[flat_nrc_sentiment$Sentiment == "negative" | flat_nrc_se
small_flat_nrc_sentiment <- filter_flat_nrc_sentiment[-c(1, 2), ]


# Plot the sentiment analysis results
filter_flat_nrc_sentiment[filter_flat_nrc_sentiment$n >= 25,] %>%
  ggplot(aes(x = factor(Word), y = n, fill = Sentiment)) +
  geom_col(position = "dodge", color = "black", linewidth = 0.2) +
  labs(title = "Flat NRC Sentiment Analysis", x = "Text ID", y = "Count") +
  scale_fill_manual(values = c("positive" = "green", "negative" = "red")) +
  theme_minimal() +
  theme(text = element_text(size = 9),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.2))
```


Flat NRC Sentiment Analysis
Count
600
400
200
0
ass
augment
bad
ban
buddy
bug
charm
deal
die
dumb
fault
forget
fun
gold
good
hit
holy
join
lose
lost
love
music
nay
patch
pivot
player
pretty
quest
quiet
radiant
real
shit
top
trash
twitch
wait
Text ID
Sentiment
🟥 negative
🟩 positive

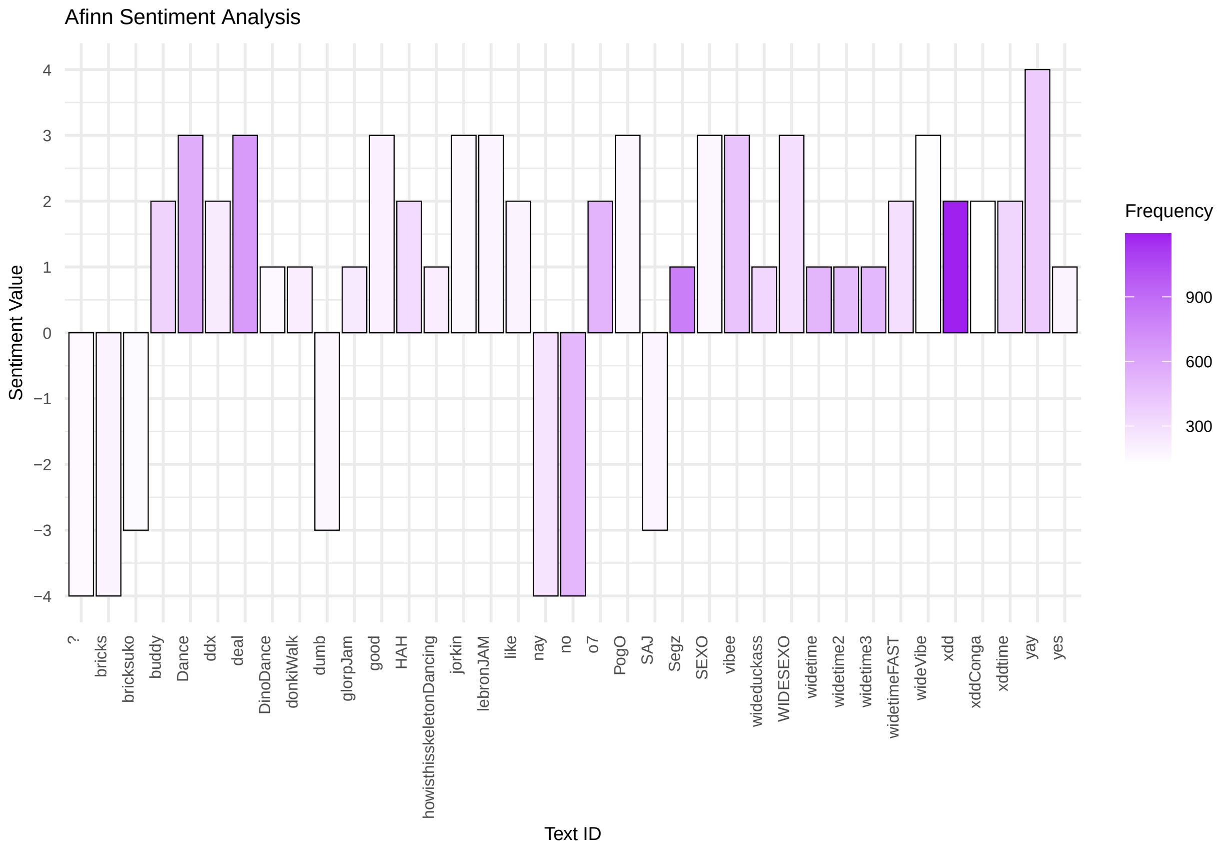
Afinn Custom Lexicon

```r
afinn_sentiment <- get_sentiments(lexicon = "afinn")
afinn_lexicon <- data.frame("Word" = afinn_sentiment$word, "Sentiment" = afinn_sentiment$value)
```

```r
afinn_twitch_words <- c("xdd", "Segz", "deal", "Dance", "o7", "widetime", "widetime3", "widetime2", "vil
afinn_twitch_words_sentiment <- c(2, 1, 3, 3, 2, 1, 1, 1, 3, 2, 2, 1, 2, 3, 2, -4, 1, 2, 1, 1, -4, -3, 3
afinn_twitch_lexicon <- data.frame("Word" = afinn_twitch_words, "Sentiment" = afinn_twitch_words_sentime

#laugh at = 2, movement = 1, positive emote = 3, positive = 4, negative emote = -3, negative = -4
afinn_new_lexicon <- rbind(afinn_lexicon, afinn_twitch_lexicon)
```

Afinn Sentiment Analysis Graph

```r
# Join with sentiment lexicon to classify words
afinn_sentiment <- split_messages %>%
  inner_join(afinn_new_lexicon, by = "Word") %>%
  count(Word, Sentiment, sort = TRUE) %>%
  ungroup()

# Filter
filter_afinn_sentiment <- afinn_sentiment[afinn_sentiment$n >= 125, ]
filter_afinn_sentiment$Sentiment[filter_afinn_sentiment$Word == "no"] <- -4

# Plot the sentiment analysis results
filter_afinn_sentiment %>%
  ggplot(mapping = aes(x = Word, y = Sentiment, fill = n)) +
  geom_col(color = "black", linewidth = 0.2) +
  theme_minimal() +
  theme(text = element_text(size = 7),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = -0.2)) +
  scale_y_continuous(label = seq(-5, 5), breaks = seq(-5, 5)) +
  labs(title = "Afinn Sentiment Analysis", x = "Text ID", y = "Sentiment Value", fill = "Frequency") +
  scale_fill_gradient(low = "white", high = "purple")
```


Afinn Sentiment Analysis
Sentiment Value
4
3
2
1
0
−1
−2
−3
−4
?
bricks
bricksuko
buddy
Dance
ddx
deal
DinoDance
donkiWalk
dumb
glorpJam
good
HAH
howisthisskeletonDancing
jorkin
lebronJAM
like
nay
no
o7
PogO
SAJ
Segz
SEXO
vibee
wideduckass
WIDESEXO
widetime
widetime2
widetime3
widetimeFAST
wideVibe
xdd
xddConga
xddtime
yay
yes
Text ID
Frequency
900
600
300