Sentiment Analysis of COVID-19 Tweets

TEAM Members: Vignesh Murugan, Vivek Satya Sai Veera Venkata Talluri, Bhuvanesh So Muruganandam, Pardhu Burlu

Group Number : 10

Date: 6th November 2024

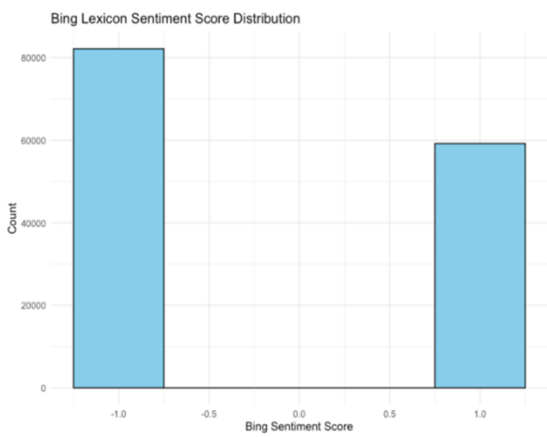
class information: OFFLINE

The COVID-19 tweets dataset, obtained from Kaggle, captures public sentiment through tweets containing keywords related to the pandemic. Following data collection, extensive preprocessing steps, including converting text to lowercase, removing URLs, mentions, hashtags, and special characters, prepared the data for sentiment analysis. Each tweet was then tokenized, with stopwords removed to focus on meaningful words. Three sentiment lexicons—Bing, AFINN, and NRC—were applied to categorize the tweets. These lexicons offer varying insights: Bing provides a binary positive/negative sentiment classification, AFINN measures sentiment intensity on a scale, and NRC identifies specific emotions like joy, anger, and sadness.

To analyze sentiment trends, positive and negative word counts were compared across the lexicons, revealing each lexicon's unique bias. Visualization techniques, including bar plots and word clouds, showcased the most common positive and negative words, with terms like "hope" and "fear" prominently featured. These analyses provide a nuanced view of public responses to COVID-19, reflecting diverse emotional reactions to the pandemic's challenges.

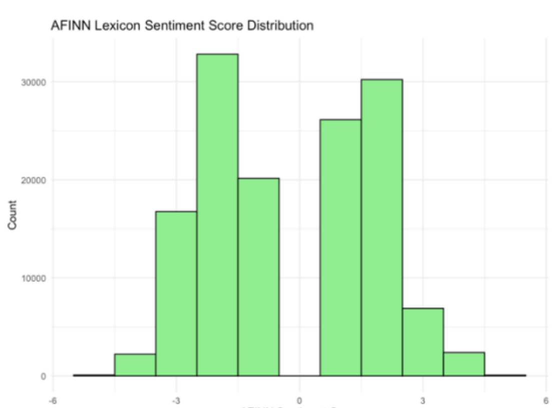
There were no missing values in text variable of the data. We are only using the text variable in the entire dataset.

PLOT 1:



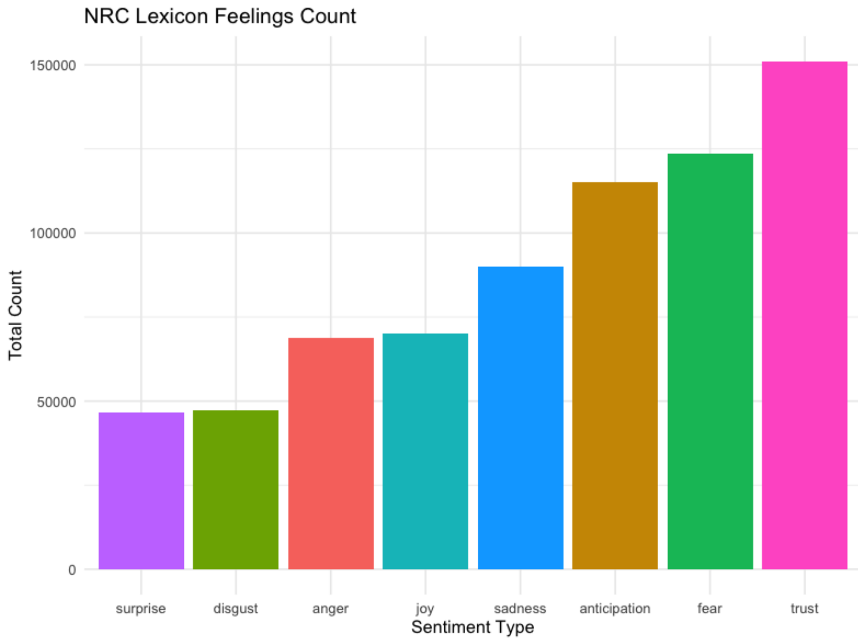
The plot shows the distribution of positive and negative sentiment words when using bing lexicon. We have more negative sentiment words compared to positive.

PLOT 2:



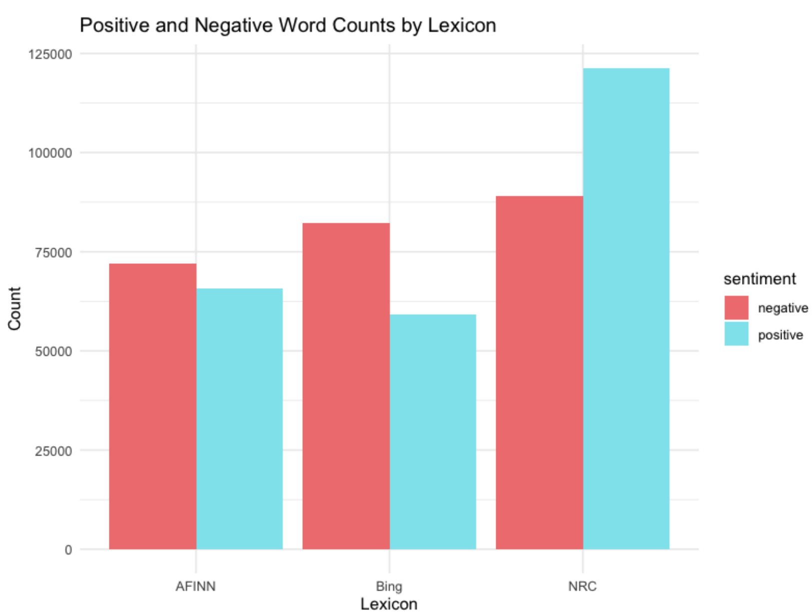
The plot shows the distribution of positive and negative sentiment words when using AFINN lexicon. It seems we have more words in sentiment values 2 and -2.

PLOT 3:



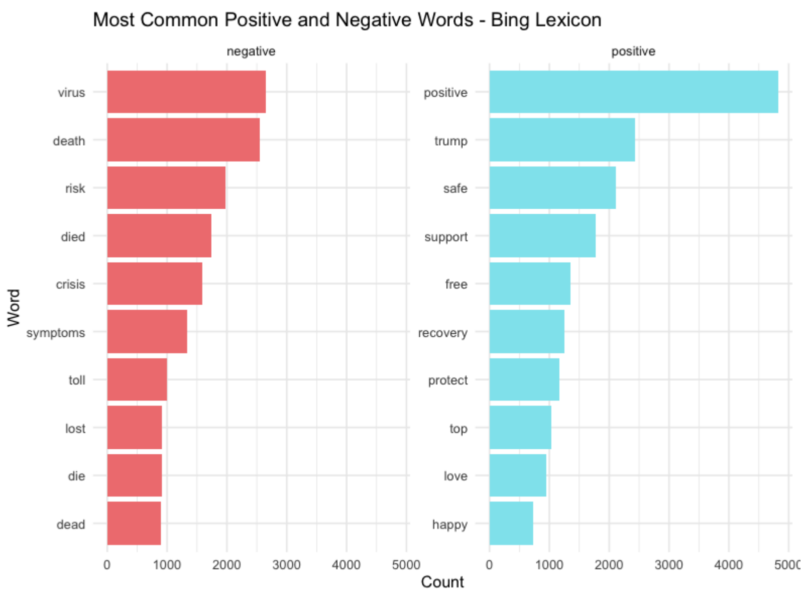
The Plot shows the different sentiments of words when using NRC lexicon.It seems like trust has the most count of words in the dataset.

Plot 4 :



The plot shows the positive and negative word counts by each lexicons. When using NRC , there more are positive words than negative, whereas when using other 2 lexicons we have more negative words.

Plot 5 :

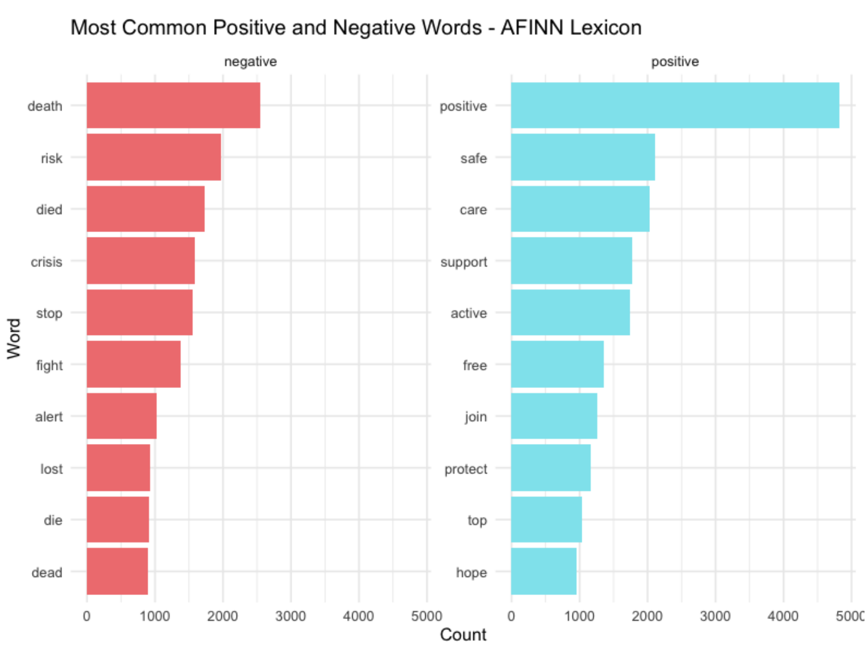


This plot shows the most common positive and negative words when using the bing lexicon.

The most 2 common +ve words are "positive", "trump" and -ve words are "virus" and "death".

In this case positive is actually a negative sentiment , the bing lexicons capturing of word positive is invalid.

Plot 6:

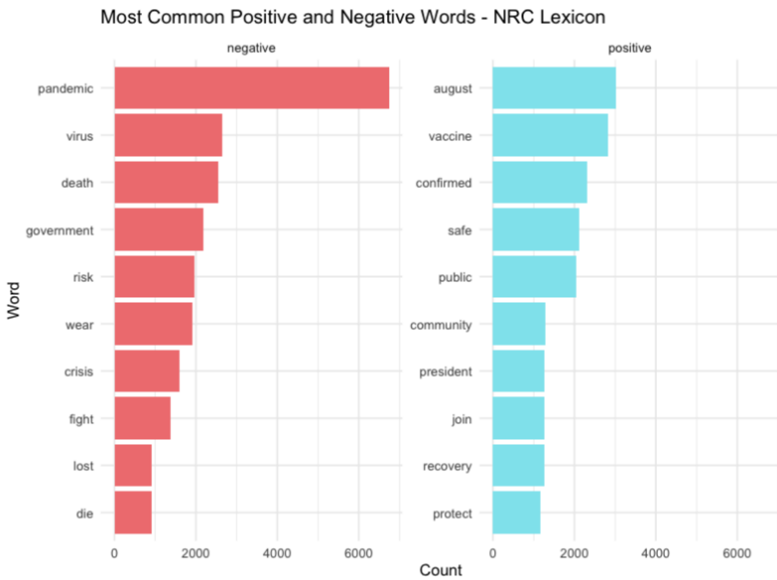


This plot shows the most common positive and negative words when using the affin lexicon.

The most 2 common +ve words are "positive", "safe" and -ve words are "death" and "risk".

In this case positive is actually a negative sentiment , the affin lexicons capturing of word positive is invalid.

Plot 7:



This plot shows the most common positive and negative words when using the NRC lexicon.

The most 2 common +ve words are "august", "vaccine" and -ve words are "pandamic" and "virus".

As of now we have completed everything we informed in the project proposals, we are planning to implement the same with tf-idf and also try to implement sentiment analysis using word embeddings if posibble.

Appendices:

```
# Load required libraries
library(tidyverse)
library(tidytext)
library(tm)
library(ggplot2)

# Load the tweets data
tweets <- read_csv("covid19_tweets.csv")

# Check for missing values in each column
missing_values <- sapply(tweets, function(x) sum(is.na(x)))
missing_values <- data.frame(Column = names(missing_values), MissingCount =
missing_values)
missing_values <- missing_values %>% filter(MissingCount > 0)
print(missing_values)

# Clean text: remove URLs, mentions, hashtags, and special characters
clean_text <- tweets %>%
  mutate(cleaned_text = map_chr(text, ~ .x %>%
                 tolower() %>%
                 str_replace_all("http\\S+|www\\S+", "") %>%  # Remove URLs
                 str_replace_all("@\\w+", "") %>%          # Remove mentions
                 str_replace_all("#\\w+", "") %>%          # Remove hashtags
                 str_replace_all("[^a-zA-Z\\s]", "")))      # Remove special characters
```

```
# Tokenize and remove stop words
tweets_tokenized <- clean_text %>%
  unnest_tokens(word, cleaned_text) %>%
  anti_join(stop_words, by = "word")

# Analyze sentiment using BING lexicon
bing_scores <- tweets_tokenized %>%
  inner_join(get_sentiments("bing"), by = "word", relationship = "many-to-many") %>%
  count(id = row_number(), sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(bing_sentiment_score = positive - negative)

# Analyze sentiment using AFINN lexicon
afinn_scores <- tweets_tokenized %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(id = row_number()) %>%
  summarise(afinn_sentiment_score = sum(value))

# Analyze sentiment using NRC lexicon
nrc_scores <- tweets_tokenized %>%
  inner_join(get_sentiments("nrc"), by = "word", relationship = "many-to-many") %>%
  count(id = row_number(), sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(
    nrc_positive_score = positive,
```

```
    nrc_negative_score = negative
  )

# View results
print(bing_scores)
print(afinn_scores)
print(nrc_scores)

###########
#  BING Lexicon Sentiment Score Distribution
###########

ggplot(bing_scores, aes(x = bing_sentiment_score)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  theme_minimal() +
  labs(
    title = "Bing Lexicon Sentiment Score Distribution",
    x = "Bing Sentiment Score",
    y = "Count"
  )

###########
#  AFINN Lexicon Sentiment Score Distribution
###########

ggplot(afinn_scores, aes(x = afinn_sentiment_score)) +
```

```
  geom_histogram(binwidth = 1, fill = "lightgreen", color = "black") +
  theme_minimal() +
  labs(
    title = "AFINN Lexicon Sentiment Score Distribution",
    x = "AFINN Sentiment Score",
    y = "Count"
  )

###########
#  NRC Lexicon Sentiment Score Distribution
###########
nrc_scores
nrc_long <- nrc_scores %>%
  pivot_longer(cols = c(nrc_positive_score, nrc_negative_score),
               names_to = "sentiment_type", values_to = "count") %>%
  pivot_longer(cols = c(disgust, anticipation, joy, trust, surprise, anger, sadness, fear),
               names_to = "Feelings", values_to = "Feelings_count")
nrc_long

# Summarize the counts by sentiment_type
nrc_summary <- nrc_long %>%
  group_by(Feelings) %>%
  summarise(total_count = sum(Feelings_count, na.rm = TRUE))

print(nrc_summary)
```

```
# Plotting the summed counts

ggplot(nrc_summary, aes(x = reorder(Feelings, total_count), y = total_count, fill = Feelings))
+

  geom_bar(stat = "identity", position = "dodge") +

  theme_minimal() +

  labs(

    title = "NRC Lexicon Feelings Count",

    x = "Sentiment Type",

    y = "Total Count"

  ) +

  theme(legend.position = "none")



###########

#  Positive and Negative words - BING

###########


bing_scores

bing_counts <- bing_scores %>%

  pivot_longer(cols = c(positive, negative), names_to = "sentiment", values_to = "count")
%>%

  group_by(sentiment) %>%

  summarise(total = sum(count))

bing_counts


###########

#  Positive and Negative words - AFINN
```

```
###########

tweets_tokenized
afinn_counts <- tweets_tokenized %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  mutate(sentiment = if_else(value > 0, "positive", "negative")) %>%
  count(sentiment) %>%
  rename(total = n)



###########
#  Positive and Negative words - NRC
###########

nrc_scores
nrc_counts <- nrc_scores %>%
  select(nrc_positive_score, nrc_negative_score) %>%
  summarise(
    positive = sum(nrc_positive_score),
    negative = sum(nrc_negative_score)
  ) %>%
  pivot_longer(cols = everything(), names_to = "sentiment", values_to = "total")

###########
#  Positive and Negative words - Combined Plot
###########
```

```
combined_counts <- bind_rows(
  bing_counts %>% mutate(lexicon = "Bing"),
  afinn_counts %>% mutate(lexicon = "AFINN"),
  nrc_counts %>% mutate(lexicon = "NRC")
)

ggplot(combined_counts, aes(x = lexicon, y = total, fill = sentiment)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(
    title = "Positive and Negative Word Counts by Lexicon",
    x = "Lexicon",
    y = "Count"
  )+
  scale_fill_manual(values = c("positive" = "cadetblue2", "negative" = "lightcoral"))

###########
# Most Common Positive and Negative Words - BING Lexicon
###########

tweets_tokenized_bing <- tweets_tokenized %>%
  inner_join(get_sentiments("bing"), by = "word", relationship = "many-to-many")

# Count most common positive and negative words
common_words_bing <- tweets_tokenized_bing %>%
```

```
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  slice_max(n = 10, order_by = n) %>%
  ungroup()

print(common_words_bing)

# Plot
ggplot(common_words_bing, aes(x = reorder(word, n), y = n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  coord_flip() +
  theme_minimal() +
  labs(
    title = "Most Common Positive and Negative Words - Bing Lexicon",
    x = "Word",
    y = "Count"
  ) +
  scale_fill_manual(values = c("positive" = "cadetblue2", "negative" = "lightcoral"))


###########
# Most Common Positive and Negative Words - AFINN Lexicon
###########

tweets_tokenized_affin <- tweets_tokenized %>%
```

```
  inner_join(get_sentiments("afinn"), by = "word") %>%
  mutate(sentiment = if_else(value > 0, "positive", "negative"))

# Count most common positive and negative words
common_words_affin <- tweets_tokenized_affin %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  slice_max(n = 10, order_by = n) %>%
  ungroup()

# Plot
ggplot(common_words_affin, aes(x = reorder(word, n), y = n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  coord_flip() +
  theme_minimal() +
  labs(
    title = "Most Common Positive and Negative Words - AFINN Lexicon",
    x = "Word",
    y = "Count"
  ) +
  scale_fill_manual(values = c("positive" = "cadetblue2", "negative" = "lightcoral"))


###########
# Most Common Positive and Negative Words - NRC Lexicon
```

```
###########

tweets_tokenized
tweets_tokenized_nrc <- tweets_tokenized %>%
  inner_join(get_sentiments("nrc"), by = "word", relationship = "many-to-many") %>%
  filter(sentiment %in% c("positive", "negative"))

# Count most common positive and negative words
common_words_nrc <- tweets_tokenized_nrc %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  slice_max(n = 10, order_by = n) %>%
  ungroup()

# Plot
ggplot(common_words_nrc, aes(x = reorder(word, n), y = n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  coord_flip() +
  theme_minimal() +
  labs(
    title = "Most Common Positive and Negative Words - NRC Lexicon",
    x = "Word",
    y = "Count"
  ) +
  scale_fill_manual(values = c("positive" = "cadetblue2", "negative" = "lightcoral"))
```

```
##############
# wordcloud
#############

##############
# wordcloud - BING
#############

library(wordcloud)

bing_words <- tweets_tokenized %>%
  inner_join(get_sentiments("bing"), by = "word", relationship = "many-to-many") %>%
  count(word, sentiment, sort = TRUE)
bing_words

# Separate positive and negative words for Bing
positive_words_bing <- bing_words %>% filter(sentiment == "positive")
negative_words_bing <- bing_words %>% filter(sentiment == "negative")

positive_words_bing
negative_words_bing

# Plot positive word cloud for Bing with a custom-positioned title
wordcloud(words = positive_words_bing$word, freq = positive_words_bing$n, max.words = 200,
```

```
        colors = brewer.pal(8, "Dark2"), random.order = FALSE)
mtext("Positive Words - Bing Sentiment", side = 3, line = -2, cex = 1.2)

# Plot negative word cloud for Bing with a custom-positioned title
wordcloud(words = negative_words_bing$word, freq = negative_words_bing$n, max.words
= 200,
        colors = brewer.pal(8, "Dark2"), random.order = FALSE)
mtext("Negative Words - Bing Sentiment", side = 3, line = -2, cex = 1.2)

##############
# wordcloud - AFINN
#############

afinn_words <- tweets_tokenized %>%
  inner_join(get_sentiments("afinn"), by = "word", relationship = "many-to-many") %>%
  mutate(sentiment = if_else(value > 0, "positive", "negative")) %>%
  count(word, sentiment, sort = TRUE)
afinn_words

# Separate positive and negative words for AFINN
positive_words_affin <- afinn_words %>% filter(sentiment == "positive")
negative_words_affin <- afinn_words %>% filter(sentiment == "negative")

# Plot positive word cloud for AFINN
wordcloud(words = positive_words_affin$word, freq = positive_words_affin$n, max.words
= 200,
        colors = brewer.pal(8, "Dark2"), random.order = FALSE)
```

```
# Plot negative word cloud for AFINN

wordcloud(words = negative_words_affin$word, freq = negative_words_affin$n, max.words
= 200,

        colors = brewer.pal(8, "Dark2"), random.order = FALSE)


##############

# wordcloud - NRC

#############


nrc_words <- tweets_tokenized %>%

  inner_join(get_sentiments("nrc"), by = "word", relationship = "many-to-many") %>%

  filter(sentiment %in% c("positive", "negative")) %>%

  count(word, sentiment, sort = TRUE)


# Separate positive and negative words for NRC

positive_words_nrc <- nrc_words %>% filter(sentiment == "positive")

negative_words_nrc <- nrc_words %>% filter(sentiment == "negative")


# Plot positive word cloud for NRC

wordcloud(words = positive_words_nrc$word, freq = positive_words_nrc$n, max.words =
200,

        colors = brewer.pal(8, "Dark2"), random.order = FALSE)


# Plot negative word cloud for NRC

wordcloud(words = negative_words_nrc$word, freq = negative_words_nrc$n, max.words =
200,
```

```
colors = brewer.pal(8, "Dark2"), random.order = FALSE)
```