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:

A graph with blue bars

Description automatically generated

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:

A graph of green bars

Description automatically generated

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.

A graph with different colored bars

Description automatically generatedPLOT 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.

A graph of different colored bars

Description automatically generated with medium confidencePlot 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.

A graph with different colored squares

Description automatically generatedPlot 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:

A graph with a bar graph

Description automatically generated with medium confidence

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.

A graph of positive and negative words

Description automatically generatedPlot 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)