Sentiment Analysis on IMDB Dataset of 50K Movie Reviews

By: Syed Norman Daniel Bin Syed Mahadir

No matrices :P125754

Analyzing movie reviews is essential for understanding audience reactions to films. Natural language processing and text analytics techniques may be used to anticipate the number of good and bad reviews when big datasets, such as IMDB's 50,000 movie reviews, are available. Before the start of sentiment analysis, preprocessing the raw text of the movie reviews is the first stage in the process. To clean and format the data, techniques such as eliminating punctuation, changing text to lowercase, and tokenization will be used. Stop word removal, stemming, or lemmatization may also be used to improve the quality of the text representation. By using this method, we want to reveal insights into the sentiment distribution within the movie review dataset, allowing us to gain a deeper understanding of audience impressions.

As we can see from the figure 1 shows the top 20 words that are frequently used in the movie review, most of them are positively good reviews. The word “movi”,”film” are the highest things that the audience love to reviews followed by “charact”,”stori” and lastly “scene”. Those words are the main topic that the audience always talks about whether there are good or bad reviews based on the word frequencies. This visualization can give useful insights into the reviewers' attitudes, views, and producers to help in gaining a better grasp of the general perspective of the films. In addition, instead of using a rectangle or spherical shape, using the text in the shape of a diamond is the best way to visualize since these two have good frequency.

Next, Figure 3, 4 and 5 show the association word term graph which contains higher than 0.14 correlation. Based on the graph 3, the word with “movi” has highest correlation with “like” which represent the audience like the movie and followed by word term “bad” which represent the audience dislike the movie. Moving in figure 4, the word term “bad”with “poorlywritten” has the highest correlation which indicates that the audience critics the poor scripts lyrics which correlate with word “bad”.

The sentiment analysis performed on the text using emotion categorization yielded exciting results. The analysis classified emotions into three categories: good, negative, and disgusting. According to the findings, the positive class had the highest occurrence of emotions, followed

by the negative class. This shows that the text has a lot of positive feelings, followed by a lot of negative sentiments. The disgust class, on the other hand, had the lowest incidence of feelings. This suggests that there were few instances of disgusting language in the text. The existence of pleasant and negative emotions at higher frequency than disgust shows that the text may elicit a variety of emotions but is less likely to elicit disgust.

In the figure 7 shows the positive word cloud which blue color displays the most often occuring positive keywords, graphically expressing their predominance. These positive words indicate the text's overall hopeful tone. The word cloud analysis reveals which positive terms are used the most frequently, allowing us to find the essential themes or concepts linked with positivity. Similarly, the negative word cloud which red color displays the most prevalent negative keywords in the text. These terms represent a negative or pessimistic emotion exhibited in the dataset.

Lastly, the sentiment analysis performed on the movie review dataset yielded useful insights into the feelings expressed by the audience. We obtained an insight of the overall favorable or negative sentiment linked with the film by analyzing the data. This data may be used to gauge audience reactions, assess the film's success, and guide film industry decision-making.

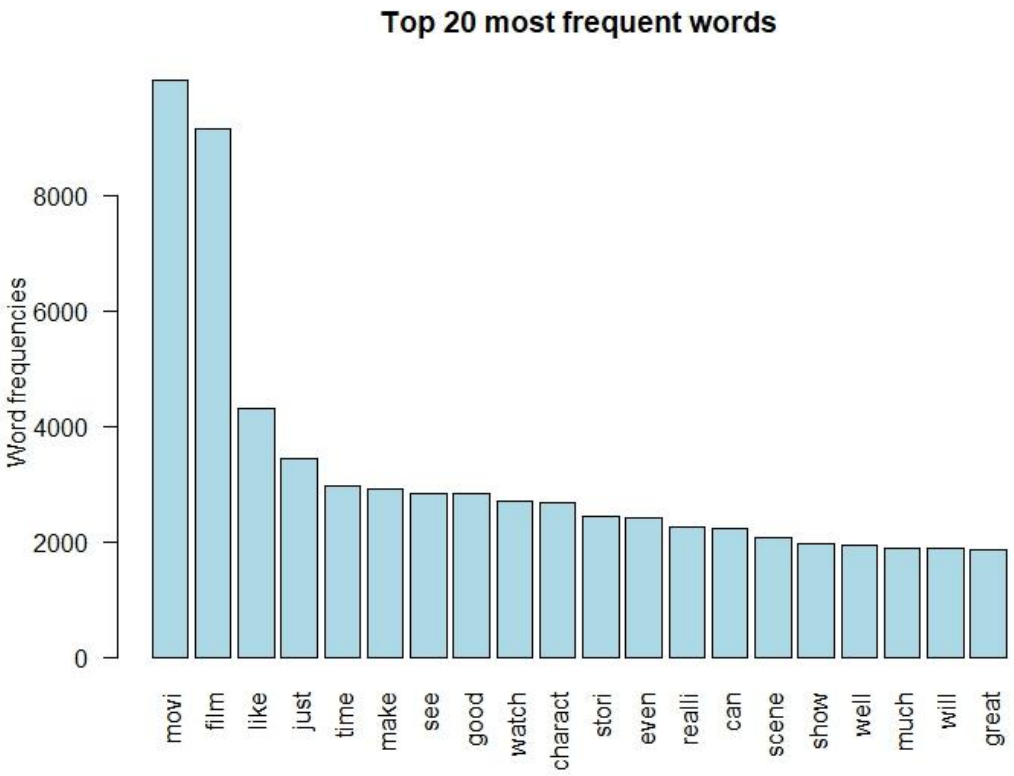

Figure 1 shows the top 20 words that audience use in reviews movie

Figure 2 shows the the word cloud in diamond shape

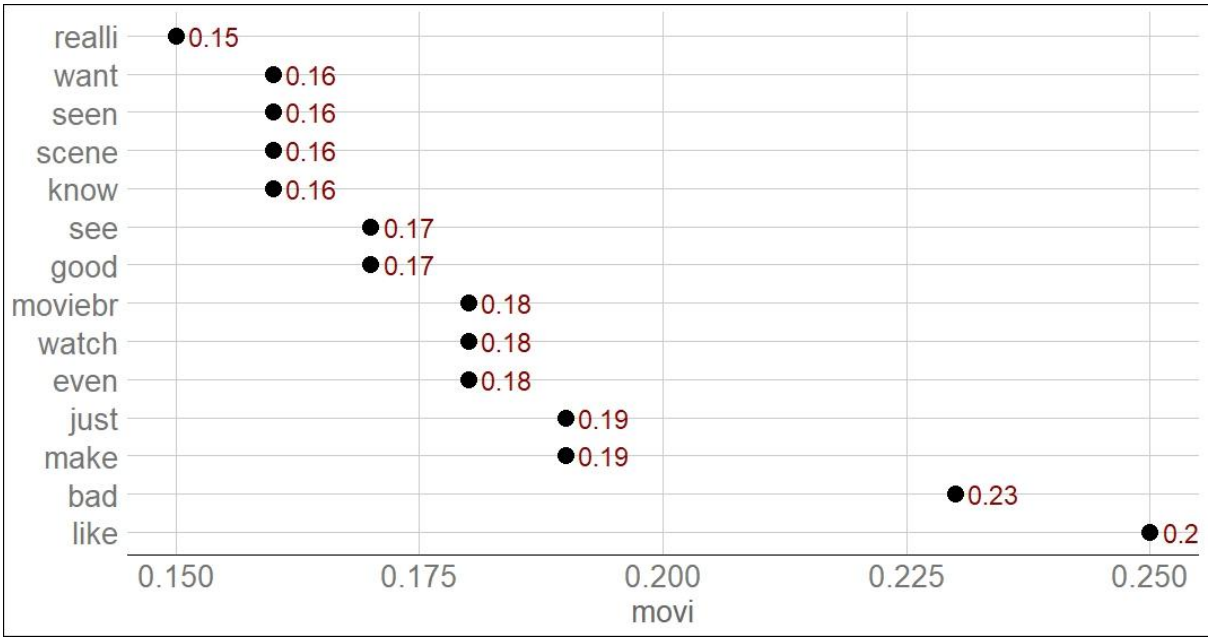

Figure 3 shows the correlation with the word “movi”

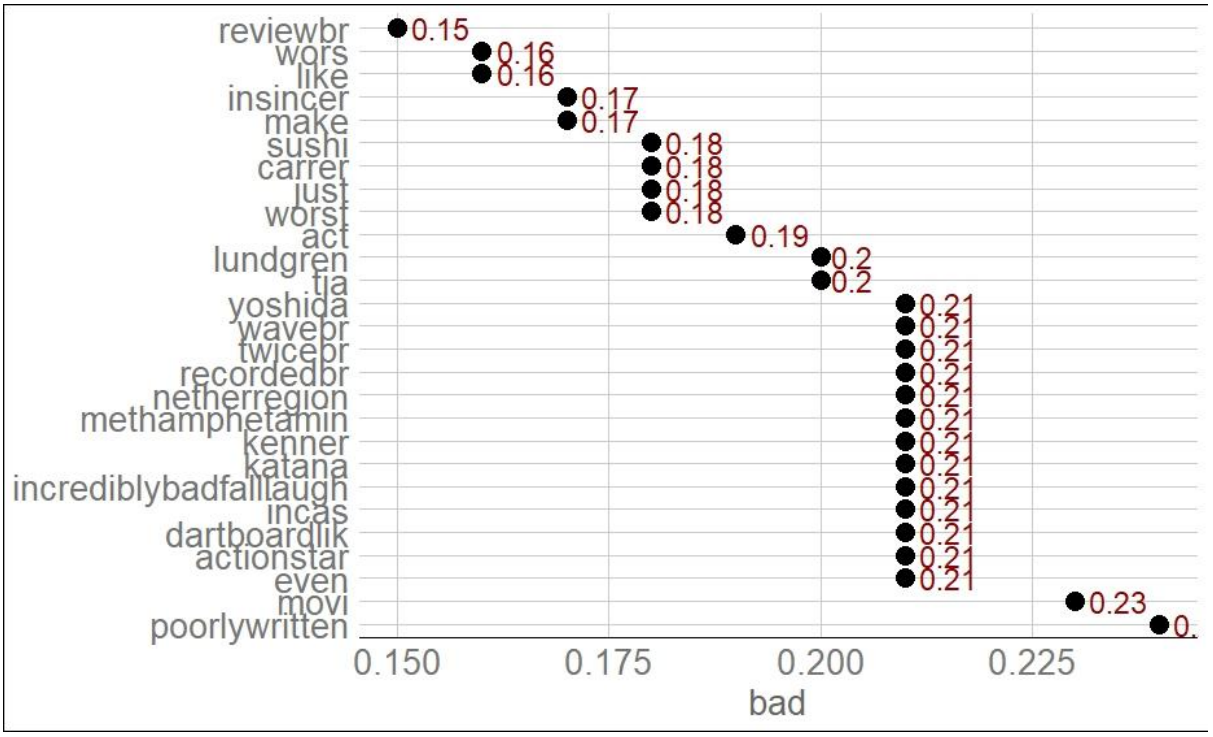

Figure 4 shows the correlation with word “bad” which represent negative word

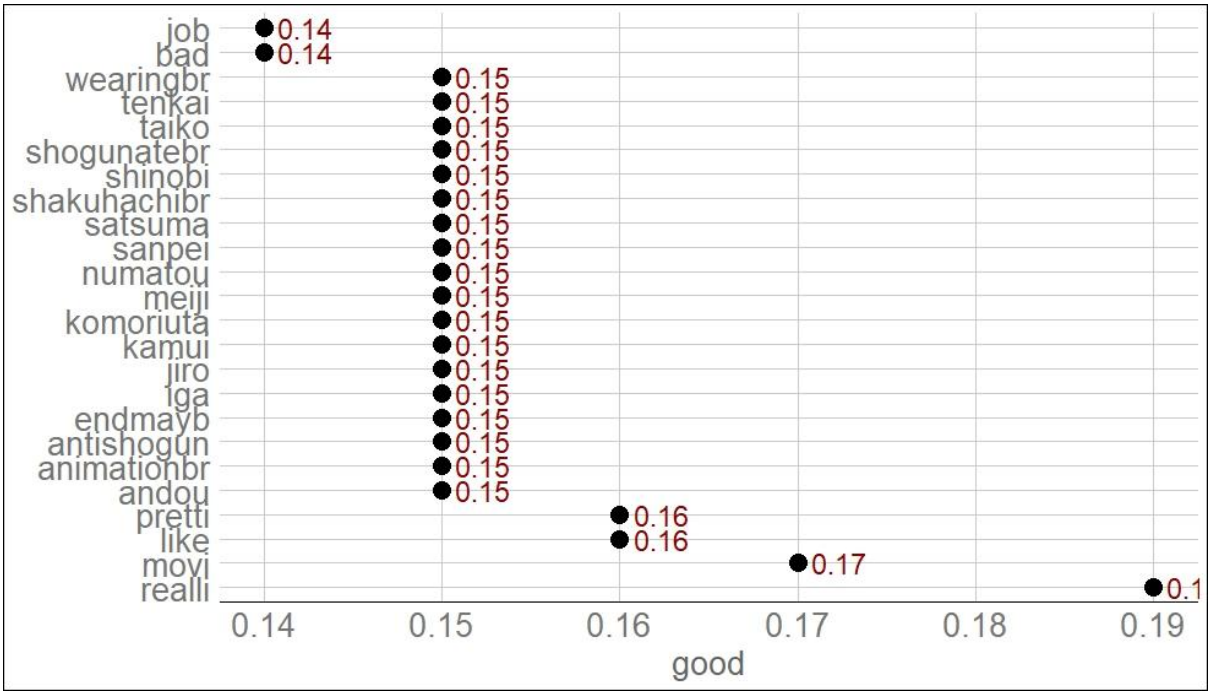

Figure 5 shows the correlation with word “good” which represent positive word

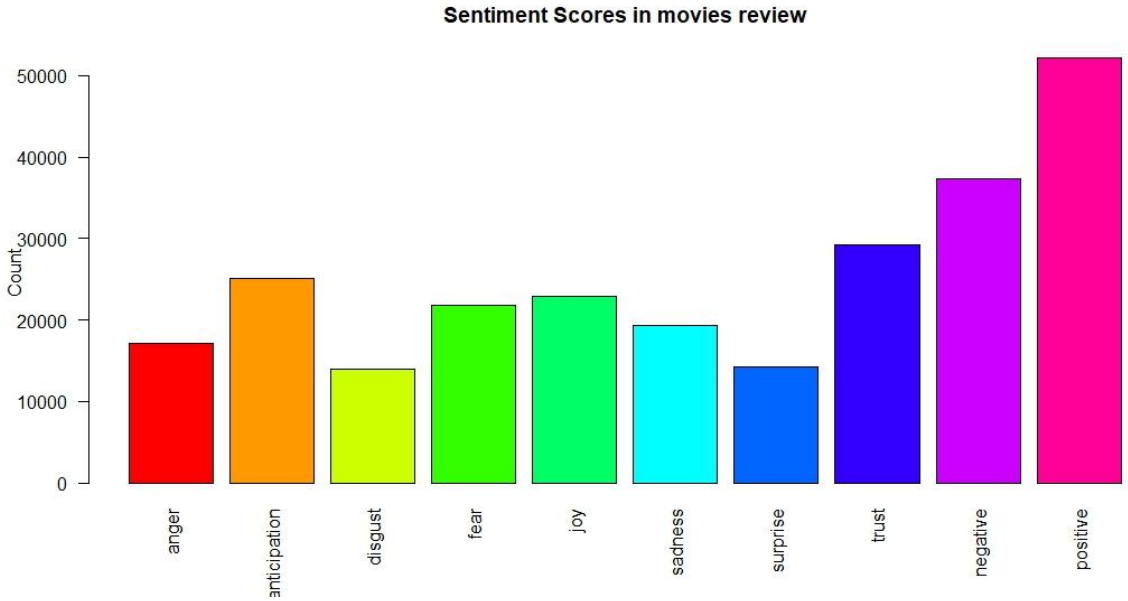

Figure 6 shows the lexicon based on the movie reviews dataset

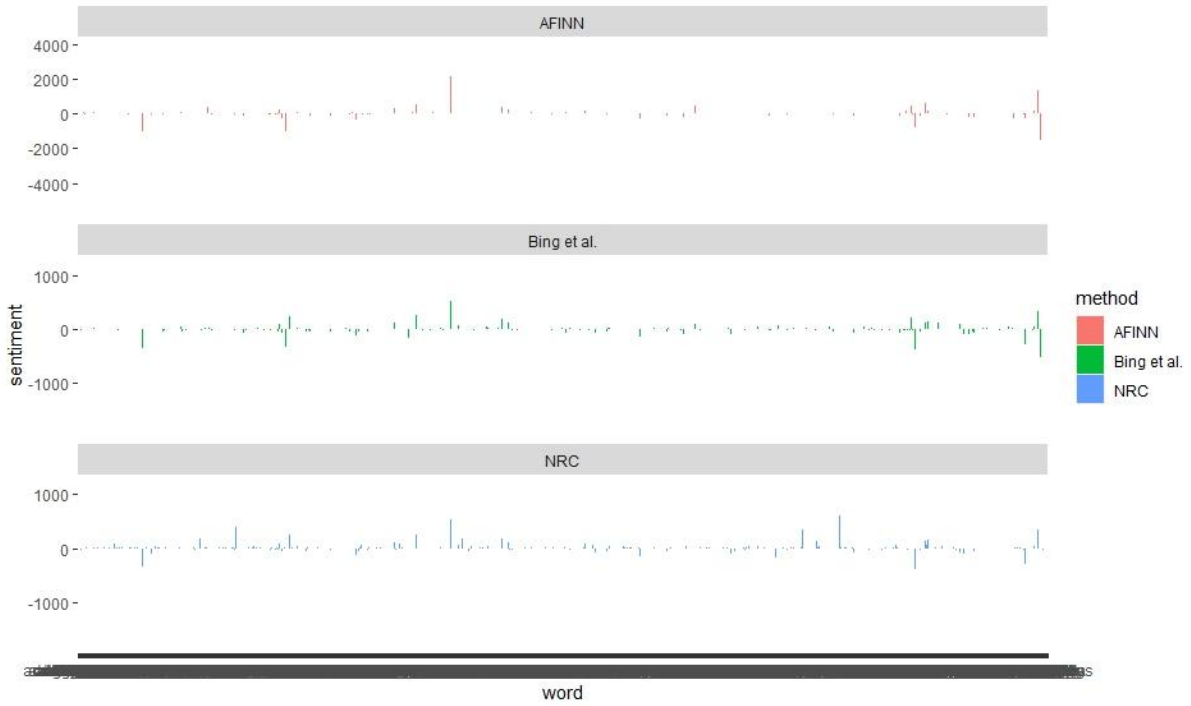

Figure 6 shows the sentiment score with different lexicons

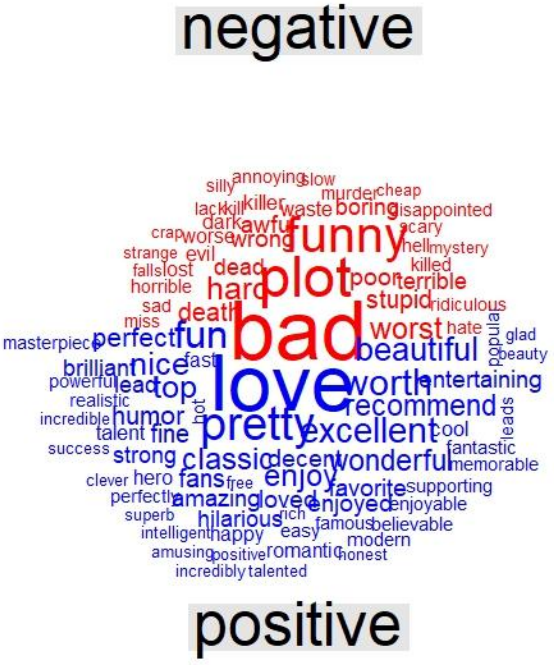

Figure 7 shows the list of positive and negative words in the review

APPENDIX

##part2

```
text=readLines(file.choose())
docs=Corpus(VectorSource(text))
inspect(docs)
toSpace=content_transformer(function(x,pattern)gsub(pattern," ",x))
docs=tm_map(docs, toSpace, "/")
docs=tm_map(docs, toSpace, "@")
docs=tm_map(docs, toSpace, "\\|")
docs=tm_map(docs,content_transformer(tolower))
docs=tm_map(docs,removeNumbers)
docs=tm_map(docs,removeWords, stopwords("english"))
docs=tm_map(docs,removeWords, c("or","else","not","one","get"))
docs=tm_map(docs,removePunctuation)
docs=tm_map(docs,stripWhitespace)
docs=tm_map(docs,stemDocument)
```

 # Build a term-document matrix

```
dtm <- TermDocumentMatrix(docs)
dtm_m <- as.matrix(dtm)
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE) # Sort by descending value of frequency
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
```

 # Plot the most frequent words

```
barplot(dtm_d[1:20,]$freq, las = 2, names.arg = dtm_d[1:20,]$word,
        col ="lightblue", main ="Top 20 most frequent words",
        ylab = "Word frequencies") + coord_flip()
#generate word cloud
library(wordcloud2)
set.seed(1234)
wordcloud2(dtm_d,
           size = 0.7,
           color = "random-dark",
           shape = 'diamond',
```

```
        rotateRatio = 0.5,
        minSize = 1)

# Word Association (movi)
A=findAssocs(dtm, terms = "movi", corlimit = 0.15) # Find associations
A=as.data.frame(A)
A$terms=row.names(A)
A$terms<-factor(A$terms,levels=A$terms)

#plot the associations (movi)
ggplot( A, aes( y = terms)) +
  geom_point( aes( x =movi ), data = A, size = 5) +
  theme_gdocs() +
  geom_text( aes( x = movi, label = movi), colour ="darkred", hjust = -0.25, size = 6) +
  theme( text = element_text( size = 20), axis.title.y = element_blank())

# Word Association (good)
C=findAssocs(dtm, terms = "good", corlimit = 0.14) # Find associations
C=as.data.frame(C)
C$terms=row.names(C)
C$terms<-factor(C$terms,levels=C$terms)

#plot the associations (good)
ggplot( C, aes( y = terms)) +
  geom_point( aes( x =good ), data = C, size = 5) +
  theme_gdocs() +
  geom_text( aes( x = good, label = good), colour ="darkred", hjust = -0.25, size = 6) +
  theme( text = element_text( size = 20), axis.title.y = element_blank())

# Word Association (bad)
B=findAssocs(dtm, terms = "bad", corlimit = 0.15) # Find associations
B=as.data.frame(B)
B$terms=row.names(B)
B$terms<-factor(B$terms,levels=B$terms)
```

```
#plot the associations (bad)
ggplot( B, aes( y = terms)) +
  geom_point( aes( x =bad ), data = B, size = 5) +
  theme_gdocs() +
  geom_text( aes( x = bad, label = bad), colour ="darkred", hjust = -0.25, size = 6) +
  theme( text = element_text( size = 20), axis.title.y = element_blank())

v
## Emotion classification
d<-get_nrc_sentiment(text)
#barplot
barplot(colSums(d),
        las = 2,
        col = rainbow(10),
        ylab = 'Count',
        main = 'Sentiment Scores in movies review')


# Load the lexicon dataset (AFINN lexicon)
lexicon <- get_sentiments("afinn")

# Load the movie review dataset (assuming it's in a CSV file)
movie_reviews <- read.csv(choose.files())

# Preprocess the text data
movie_reviews <- movie_reviews %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)

# Perform sentiment analysis using the lexicon
sentiment_scores <- movie_reviews %>%
  inner_join(lexicon) %>%
  group_by(word) %>%
```

```
  summarize(sentiment_score = sum(value))

# Classify sentiment based on the sentiment scores
sentiment_scores <- sentiment_scores %>%
  mutate(sentiment_label = ifelse(sentiment_score >= 0, "Positive", "Negative"))

# Calculate the percentage of positive and negative reviews
sentiment_summary <- sentiment_scores %>%
  summarize(positive_reviews = sum(sentiment_label == "Positive"),
            negative_reviews = sum(sentiment_label == "Negative"),
            total_reviews = n(),
            positive_percentage = positive_reviews / total_reviews * 100,
            negative_percentage = negative_reviews / total_reviews * 100)

# Print the sentiment summary
print(sentiment_summary)

#Compare three different lexicons
afinn <- movie_reviews %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(word) %>%
  summarize(sentiment= sum(value))%>%
  mutate(method = "AFINN")

bing_and_nrc <- bind_rows(
  movie_reviews %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  movie_reviews %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
    mutate(method = "NRC")) %>%
```

```
  count(method, word, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)

bind_rows(afinn,
          bing_and_nrc) %>%
  ungroup()  %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_bar(alpha = 0.8, stat = "identity", show.legend = FALSE) +
  facet_grid(word ~ method)
plot<-bind_rows(afinn,
          bing_and_nrc) %>%
  ggplot(aes(word, sentiment, fill = method)) +
  geom_col(show.legend = TRUE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")

ggsave("plot.png", plot, width = 10, height = 8)
```

#Most common positive and negative words

#show most bad and good using word cloud

```
movie_reviews %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "blue"),
                   max.words = 100)
```