IMDB REVIEWS ANALYSIS

Top 5 Rated & Bottom 5 Rated Movies' Reviews

Tuba Ali

Hult Business School 2020

1. Basic Information About the Analysis

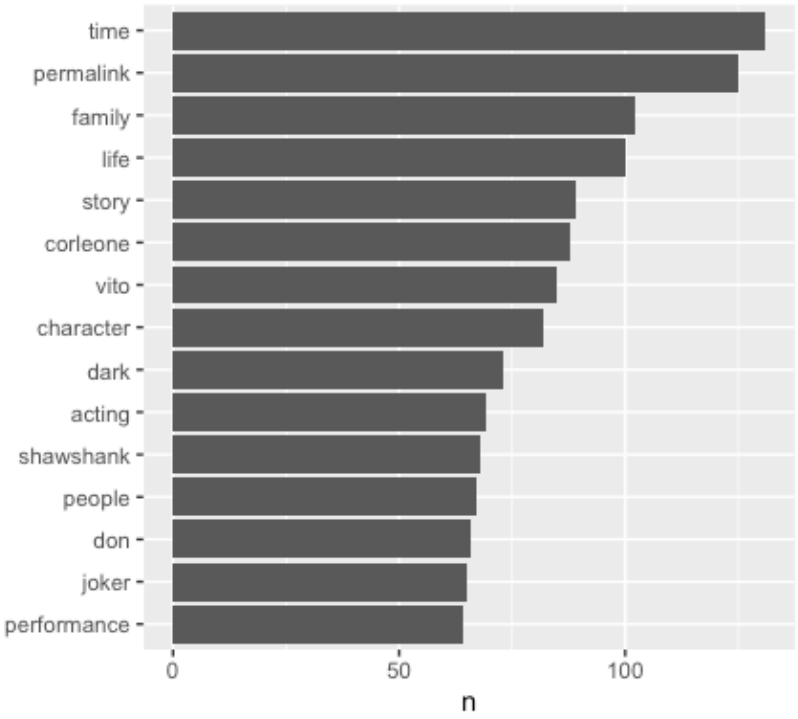
In this case I try to analyze wording that people use when they review “Good” and “Bad” Movies. Are they express themselves differently? Which words are they choosing? Is there similarities between good reviews and bad reviews? For analysis, I took Top Rated 5 movies’ Reviews and Bottom Rated 5 movies’ Reviews from IMDB. Firstly, I analyze them separately then I gathered information together.

Top 5 Rated movies were:
The Shawshank Redemption 1994),
The Godfather (1972), The Dark Knight (2008),
The Godfather: Part II (1974),
The Lord of the Rings: The Return of the King (2003).

Bottom 5 Rated movies were:
Saving Chrismas (2014),
Code name: K.O.Z. (2015),
Turks in Space (2006),
Going Overboard (1989).

2. Analysis Top 5 Movies

To understand data, I tokenized the both text Top 5 Movies’ Review and Bottom 5 Movies’ Review separately. After tokenizing the data, I realize there is couple words that repeated and not giving any business inside. I created custom stop words for Top 5 after checking out tokenized data. Because of the website, “film”, “movie”, “Movies”, “10”, “review”, “vote”, “sign” was repeatedly used. Additional to website wording I also anti join the movie names and couple character names. Also, I realize word “helpful” repeating so I check out the text document to find where word helpful used and find out it’s also one of the website words so added it to customized stop words.
Here is Top 5 tokens with customized stop words and plot of top 15 tokens:




```
> top_5_token_cust
# A tibble: 5,918 x 2
   word          n
   <chr>     <int>
 1 time        131
 2 permalink   125
 3 family      102
 4 life        100
 5 story        89
 6 corleone     88
 7 vito         85
 8 character    82
 9 dark         73
10 acting       69
# … with 5,908 more rows
```

As you can see in the plot, most used words are generally defining the story of the movies. Because of the Godfather, we can see family, Corleone, don repeating a lot. Also because of Dark Knight Rises, we see dark and joker.

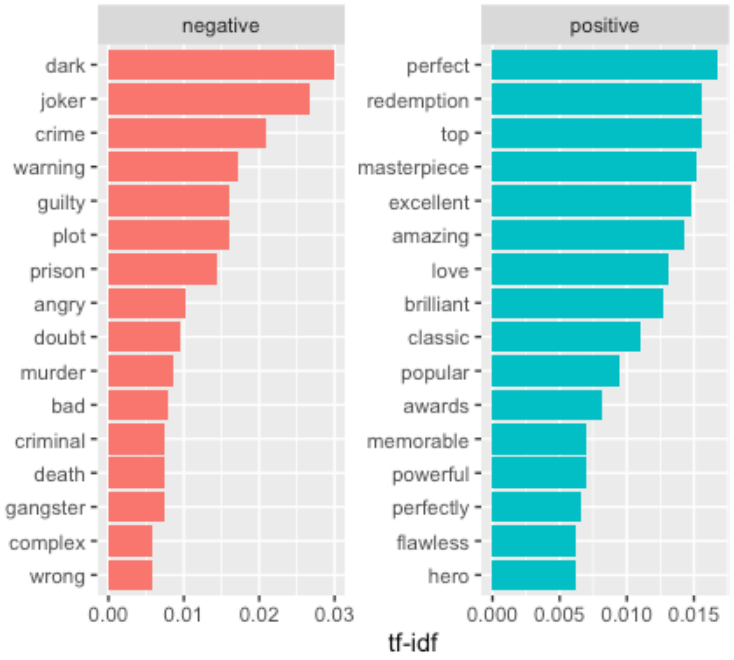
2.2 Sentiment Analysis for Top 5 Movies

For understand the reviews are positive or negative I decided to apply sentiment analysis. Rather than emotions of the “nrc” or amount of positivity or negativity from “afinn” I want to have general idea about positive-negative analyzing the reviews. So, I decided to use “bing” to find out sentiment. However, it was really hard to understand positivity-negativity of the data by checking out from result. As you can see because of most of the reviews was discussing the stories of the movies and movies was generally about the crime or crime families it categorized as negative sentiment.

```
> top_5_sentiment
# A tibble: 1,091 x 3
   word       sentiment     n
   <chr>      <chr>     <int>
 1 dark       negative     73
 2 joker      negative     65
 3 crime      negative     51
 4 warning    negative     42
 5 perfect    positive     41
 6 guilty     negative     39
 7 plot       negative     39
 8 redemption positive     38
 9 top        positive     38
10 masterpiece positive     37
# … with 1,081 more rows
```

2.3 Tf-Idf for Sentiment of the Top 5 Movies

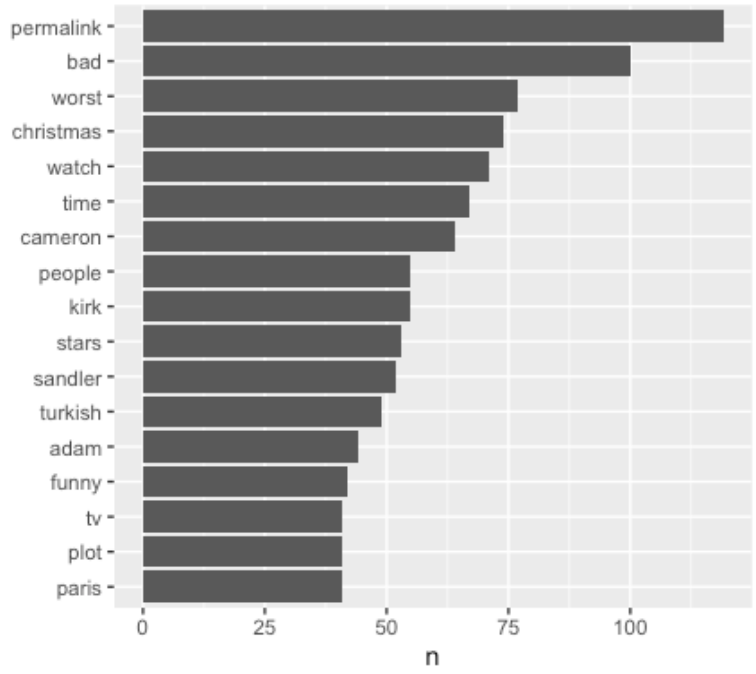
For understand the sentiment analysis better, I applied tf-idf and categorized on the sentiment. Basically, it shows positive and negative sentiments in different tables and brief effect of the words on the review. I visualized the solution by using plot:



As you see in the plot when we categorized the top 5 movies as positive negative, all the negative tokens are related to story of the movies. However, when we check out the positive side, they are basically show reviewer's opinion about the movies. "Perfect", "Masterpiece", "Classic" etc.

3.1 Analysis Bottom 5 Movies

When we move to Bottom 5 Movies' Reviews, I also followed same steps as I applied for Top 5. I created customized stop words, after I saw the tokens. In Bottom 5 custom stop words I only included website repeat words. Here is the frequency of tokens of Bottom 5 Movies' Reviews:




```
> bottom_5_token_cust
# A tibble: 3,985 x 2
   word          n
   <chr>     <int>
 1 permalink   119
 2 bad         100
 3 worst        77
 4 christmas    74
 5 watch        71
 6 time         67
 7 cameron      64
 8 kirk         55
 9 people       55
10 stars        53
# … with 3,975 more rows
```

Most frequent tokens are including actor names such as “Adam Sandler”, were is movie originated such as “Turkish” and others. For understand the positivity-negativity of the reviews I applied sentiment analysis and use “nrc” sentiments. Here the “nrc” sentiment analysis for the bottom 5 movies.

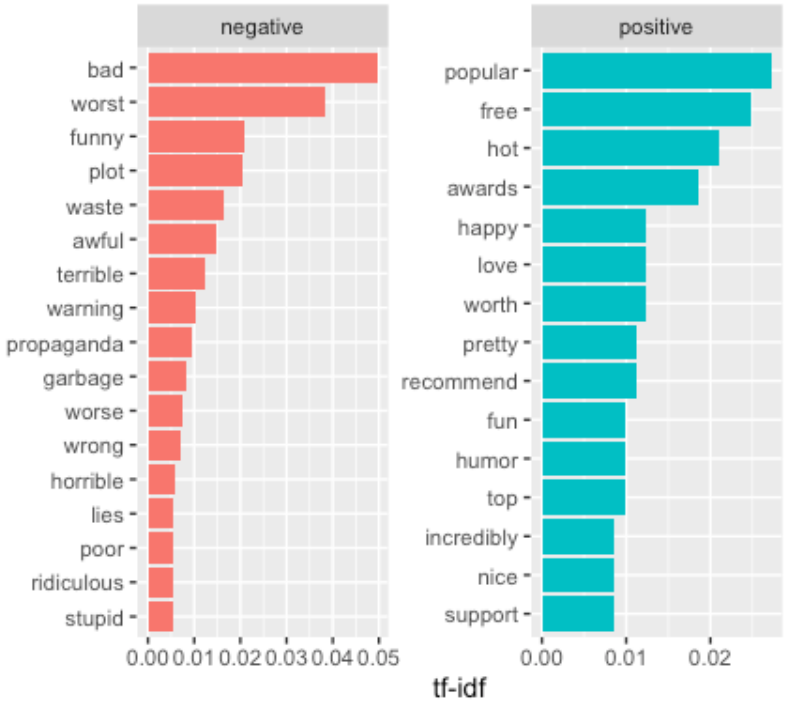
```
> bottom_5_sentiment
# A tibble: 752 x 3
   word      sentiment     n
   <chr>     <chr>     <int>
 1 bad       negative    100
 2 worst     negative     77
 3 funny     negative     42
 4 plot      negative     41
 5 waste     negative     33
 6 awful     negative     30
 7 terrible  negative     25
 8 popular   positive     22
 9 warning   negative     21
10 free      positive     20
# … with 742 more rows
```

3.3 Sentiment Analysis for Bottom 5 Movies

As you can see it’s obvious this is the bottom 5 movies because top 10 tokens are categorized as negative and they all definitive words such as “bad”, “worst”, “waste”, “terrible”. So if you don’t know which one is the bottom rated and top rated you can still understand from the sentiment analysis.

3.4 Tf-Idf for Sentiments to Bottom 5 Movies

I applied the sentiment analysis to the tf-idf for understand effect of the tokens. I gathered them in to a plot:



As you can see in the plot, Bottom 5 Rated Movies' Reviews has definitive wording in the negative side.

4. Conclusion

As we see in the Top 5 Movies people most likely to mention about story of the movie and characters. However, in bottom 5 I didn't see any clue about story telling. I related this with people bond with epic movies and they watch them couple times. They effect from the story and connect with the characters and this cause them to mention about the story.

Also, when I applied sentiment analysis to top 5 movies, I realize most of the negative words related the story so we cannot categorize them as a negative word. However, in bottom 5 I saw lot of definitive words in negative category of the sentiment analysis.

I was interested to see which words people use when they define "good" and "bad". It was interesting to see in good reviews people try to tell why they like the movie, why they attached it. In the other hand, bad reviews mostly include criticism and definitive words about how bad movie is.

5. R-Code

```
################################################################################
#
#                          ANALYZING IMDB TOP 5 & BOTTOM 5 MOVIES' REVIEWS
#
################################################################################
#Calling libraries
library(dplyr)
library(tidytext)
library(textreadr)
library(reshape2)
library(tidyr)
library(stringr)
library(wordcloud)
library(ggplot2)
library(reshape2)


################################################################################
#                              DEFINING DATA FRAMES
################################################################################
# Set working directory to where the text file is
setwd("~Users/tubaali/Desktop/Text Analytics")

# Reading text files
#IMDB Top 5 Movies Reviews
top_5 <- read_document(file = "Top 5 .txt")
top_5_df <- tibble(text =as.character(top_5))

#IMDB Bottom 5 Movies Reviews
bottom_5 <- read_document(file = "Bottom 5.txt")
bottom_5_df <- tibble(text =as.character(bottom_5))

################################################################################
#                               STOP WORDS
################################################################################
#Calling the Stop Words
data("stop_words")

#Custom Stop Words +added helpful, sign, vote, reviews bec. it's website thing
cust_stop_top<- tibble(word = c("film", "movie", "is","movies", "spoilers", "makes", "de", "ii",
"10", "review", "found", "films",
                        "godfather", "michael", "batman", "found", "1", "helpful", "sign", "vote",
"reviews"))
```

```
cust_stop_bottom <-tibble(word = c("movie", "film", "10", "1", "vote", "review", "found",
"imdb", "films", "movies", "helpful",
                          "sign", "vote", "reviews"))

#############################################################################
#                                TOKENIZING TOP 5
#############################################################################
top_5_token <- top_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)

top_5_token

#Taking out the customize stop words --> film, movie, is, spoilers etc.
top_5_token_cust <- top_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop_top) %>%
  count(word, sort = TRUE)

top_5_token_cust

#Visualize top 15 words for Top 5
top_5_token_cust %>%
  mutate(word = reorder(word, n)) %>%
  top_n(15) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()

#############################################################################
#                                SENTIMENT ANALYSIS FOR TOP 5
#############################################################################
#Get sentiments from bing
top_5_sentiment <- top_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop_top) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()
```

```
top_5_sentiment

#Making a comparrison cloud with sentiment bing --> Positive/Negative
top_5_token_cust %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("gray80", "gray20"),
                   max.words=50,
                   scale = c(0.5,0.5),
                   title.size = 1)


################################################################################
#                                SENTIMENT TF-IDF FOR TOP 5
################################################################################
#######################################td-idf to sentiment
top_5_sentiment <- top_5_sentiment %>%
  bind_tf_idf(word, sentiment, n) %>%
  arrange(desc(tf_idf))

top_5_sentiment

#Visualizing Tf-idf to bing sentiment
top_5_sentiment %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(sentiment) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~sentiment, ncol = 2, scales = "free") +
  coord_flip()

################################################################################
#                                SENTIMENT BIGRAM FOR TOP 5
################################################################################
#bigram with stop words
top_5_bigrams <- top_5_df %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
  count(bigram, sort = TRUE)
```

```
top_5_bigrams

#Bigram filtered stop words
top_5_bigrams_separated <- top_5_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

top_5_bigrams_filtered <- top_5_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

top_5_bigrams_filtered

#New bigram counts
top_5_bigram_counts <- top_5_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

top_5_bigram_counts

################################################################################
################################################################################
#                                SECOND PART
#                               Bottom 5 Analysis
################################################################################
################################################################################


################################################################################
#                               TOKENIZING BOTTOM 5
################################################################################
#Tokenizing
bottom_5_token <- bottom_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE)

bottom_5_token

#Taking out the customize stop words --> film, movie, is, spoilers etc.
bottom_5_token_cust <- bottom_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop_bottom) %>%
  count(word, sort = TRUE)
```

```
bottom_5_token_cust

#Visualize top 15 words for bottom 5
bottom_5_token_cust %>%
  mutate(word = reorder(word, n)) %>%
  top_n(15) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()

################################################################################
#                              SENTIMENT ANALYSIS FOR BOTTOM 5
################################################################################
#Adding sentiment bing
bottom_5_sentiment <- bottom_5_df %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(cust_stop_bottom) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bottom_5_sentiment

#Making a comparrison cloud with sentiment bing --> Positive/Negative
bottom_5_token_cust %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("gray80", "gray20"),
                   max.words=50,
                   scale = c(0.5,0.5),
                   title.size = 1)

################################################################################
#                              SENTIMENT TF-IDF FOR BOTTOM 5
################################################################################
#td-idf to sentiment
bottom_5_sentiment <- bottom_5_sentiment %>%
  bind_tf_idf(word, sentiment, n) %>%
  arrange(desc(tf_idf))
```

```
bottom_5_sentiment

#Visualizing Tf-idf to bing sentiment
bottom_5_sentiment %>%
 arrange(desc(tf_idf)) %>%
 mutate(word = factor(word, levels = rev(unique(word)))) %>%
 group_by(sentiment) %>%
 top_n(15) %>%
 ungroup %>%
 ggplot(aes(word, tf_idf, fill = sentiment)) +
 geom_col(show.legend = FALSE) +
 labs(x = NULL, y = "tf-idf") +
 facet_wrap(~sentiment, ncol = 2, scales = "free") +
 coord_flip()

################################################################################
#                                SENTIMENT BIGRAM FOR BOTTOM 5
################################################################################
#bottom bigram with stop words
bottom_5_bigrams <- bottom_5_df %>%
 unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
 count(bigram, sort = TRUE)

bottom_5_bigrams

#Bigram filtered stop words
bottom_5_bigrams_separated <- bottom_5_bigrams %>%
 separate(bigram, c("word1", "word2"), sep = " ")

bottom_5_bigrams_filtered <- bottom_5_bigrams_separated %>%
 filter(!word1 %in% stop_words$word) %>%
 filter(!word2 %in% stop_words$word)

bottom_5_bigrams_filtered

# new bigram counts:
bottom_5_bigram_counts <- bottom_5_bigrams_filtered %>%
 count(word1, word2, sort = TRUE)

bottom_5_bigram_counts
```