INITIAL PROJECT REPORT

SENTIMENT ANALYSIS OF MOVIE REVIEWS IN TERMS OF POLARITY, RATING AND SUBJECTIVITY

**TEAM MEMBERS**

Thanate Banchonhattakij

Poorani Jagadeesan

Yashwanth Gangapuram

**BACKGROUND**

Effective analysis of a large dataset of text-based movie reviews provide insight into the qualitative and quantitative evaluation of a movie’s success in terms of the tone of reviews, sentiment and rating (Goyal & Parulekar, 2015). Sentiment analysis involves the use of systematic identification, extraction and study of textual subjective information for gauging the general reaction, evaluating emotions and advantageous decision making in regards to making more successful movies (Wikipedia, 2020a).

Text mining algorithms and techniques allow detection of emotions in text and predict the positive or negative nature of the words helping determine the quality of the sentiment expressed in the movie review (Wikipedia, 2020b).

After initial deliberation on the nature of text needed to analyze sentiment in terms of polarity, rating and subjectivity, the IMDB movie dataset with 1000 positive reviews and 1000 negative reviews was selected for sentiment analysis (Pang & Lee, 2004).

In this project, steps are taken to understand the effects of affecting words in influencing the polarity of a movie review based on Naïve Bayes Probabilistic algorithm for text mining. Furthermore, rule-based sentiment analysis is applied to the dataset for understanding the frequency and tone of the language to better grasp the emotion expressed in the movie review.

**INTRODUCTION**

A brief discussion of the text mining techniques Naïve Bayes and rule-based sentiment analysis used for movie review analysis are given below.

**NAÏVE BAYES ALGORITHM**

The most popular text classification algorithm based on probabilistic Bayesian methods is the Naïve Bayes model and it performs well with small or large data due its relatively simple formulation (Lantz, 2015).

BAYES THEOREM

The probabilistic theorem states that given the probability of an event B occurring, the probability of an event A occurring based on the number of trails in which A occurs together with B is given by the following mathematical formulation (Wikipedia, 2020c).

Naïve Bayes assumes that each of the features present in the data set are class-conditionally independent and equally important, and even when it is not, so obtains a prediction with estimated probability.

PROBABILISTIC MODEL

The Naive Bayes classification algorithm can be mathematically represented as

The above equation states that the probability for class C of level L, given the observation of the features from F1 to Fn is the product of conditional probability on each level to the prior probability and scaling factor (1/Z).

An example likelihood table (from the frequency table) showing the appearance of 5 frequent words (features) in 100 movie reviews (divided into positive and negative class variables) are given below for each positive and negative class:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Good | | Bad | | Success | | Poor | |  |
| Likely | Yes | No | Yes | No | Yes | No | Yes | No | Tot |
| Pos | 35/50 | 15/50 | 6/50 | 44/50 | 32/50 | 18/50 | 4/50 | 46/50 | 50 |
| Neg | 2/50 | 48/50 | 43/50 | 7/50 | 13/50 | 37/50 | 33/50 | 17/50 | 50 |
| Total | 37/100 | 63/100 | 49/100 | 51/100 | 45/100 | 55/100 | 37/100 | 63/100 | 100 |

LAPLACE SMOOTHING

When an event never happens for one or more levels of a class, the probability of it occurring becomes zero, and from the above equation causes an error due to multiplication with 0, nullifying the probability of all the other features in the class of that level.

This problem can be rectified by introducing a Laplace estimator, that will add a small number to each of the counts in the frequency table, to prevent the zero probability from occurring. Generally, for large training datasets, the Laplace estimator is set to 1 so that the class-feature combination is present in the level at least once.

**SENTIMENT ANALYSIS**

Sentiment analysis on textual data is performed for polarity analysis, frequency analysis, and sentiment tone analysis. Out of the several types of sentiment analysis like fine-grained, rule-based and aspect-based techniques, the manual rule-based sentiment analysis is considered for frequency analysis of emotions in the movie reviews (Monkeylearn, 2020).

RULE-BASED LEARNING

Extracting useful information from large textual dataset employs a method of creating a bag-of-words called corpus. Rule-based learning makes sense of the variety of emotions expressed frequently and how it affects the general and overall polarity of the movie review.

Crucial to any data mining analysis is data preparation and cleaning, especially true for text-based sentiment mining. The ‘tidytext’ package provided several functionalities for the above-mentioned tasks (Gabriela et al, 2020). Tibble, an advanced data frame structure in R, is created that contains each word by movie review by file for effective sentiment analysis (Grolemund & Wickham, 2017).

Equipped the prepared data, manual-rule building is performed based on the most frequent words in the movie reviews that attribute to a pre-existing exhaustive lexicon with binary classification of emotional words (Boehmke, n.d.-b).

The two lexicons used for the purpose of sentiment analysis in this project are called ‘nrc’ and ‘bing’. The ‘nrc’ lexicon splits words into 10 categories of emotions, namely, positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust with a yes or no connotation (Mohammad, 2016).

The ‘bing’ lexicon is another English language sentiment lexicon that classifies words into positive or negative based on emotion in a binary fashion (Hu et al, 2004).

**CHALLENGE RESOLUTION**

**Data preparation challenges:** Due to the immensity and range of the textual data present in the polarity review dataset, cleaning the dataset thoroughly with respect to white spaces, numbers and stop words. This was addressed using the tidytext package functionalities for stripping the data o unwanted words (Silge, n.d.).

Some data processing and format difficulties presented itself, like converting to DocumentTermMatix and tibbles for easier analysis. This issue was tackled by understanding the different types of data frames available in R and accessing the packages that allow for conversion and handling of text datasets.

**Improving the model performance:** Adding the Laplace estimator or changing the stopwords in the naïve Bayes classifier algorithm to include more custom words, did not present significant model improvement in terms of the false positive and false negative values.

In rule-based sentiment analysis, on the other hand, the addition of anti\_join to remove stop words produced a significant word frequency count decrease in the final analysis (Silge, n.d.).

Understanding the reason for using right\_join or inner\_join while combining the ‘nrc’ and ‘bing lexicons proved puzzling, but after careful study of the function and the results of joining two tables in the R environment offered some clarity (Wickham, n.d.).

**Visualizing the results:** Generating meaningful reports and creative visualizations of the algorithm results and their implications proved challenging and will be explored more in the coming week as the sentiment analysis is still in progress.

**IMPLEMENTATION**

**NAÏVE BAYES**

DATA PROCESSING

1. Import necessary packages, set directory and load the text files unto the R environment using readtext() function (Benoit et al, 2020).
2. Create class label type for positive and negative reviews, combine the data into one data frame and randomize the columns.
3. Consolidate the text files into single corpus, a collection of data in text essential for analysis (Feinerer, n.d.).
4. Perform several data cleaning techniques to make the corpus useful for sentiment analysis.
   1. Convert to lowercase.
   2. Remove numbers.
   3. Remove Stopwords.
   4. Remove punctuations.
   5. Convert words to root form (stemming).
   6. Strip white space.
5. Transform the data to DocumentTermMatrix form.

TRAINING THE MODEL

1. Split and create training (75% - 1500 reviews) and testing data (25% - 500 reviews) (Boehmke, n.d.-a).
2. Create labels for positive and negative reviews for later validation.
3. Perform frequent word search and indicate the features for training the model.
4. Build the Naïve Bayes Classifier model for the training data (Meyer & Xue, n.d.).

EVALUATION

1. Use the classifier model to predict the polarity in the testing data and obtain CrossTable.

PERFORMANCE IMPROVEMENT

1. Change the Stopwords to include more words along with the default set.
2. Change the frequency of the indicator features for better training.
3. Apply the Laplace estimator and compare the results before and after Laplace smoothing.

**RULE-BASED SENTIMENT ANALYSIS**

DATA PROCESSING

1. Import necessary packages, set directory and load the text files unto the R environment using readtext() function (Benoit et al, 2020).

Gather the positive and negative movie review data files into one-token-per-row using the unnest\_tokens() function for converting the data frame into a text column (Silge & Robinson, 2020). This aids in easier sentiment analysis.

1. Combine the positive and negative review tibbles into a single object using rbind() function(R-core, n.d.).

TRAINING THE MODEL – NRC, BING

1. Filter the sentiments (each word representing a particular emotion as expressed in the lexicon) from the movie review tibble.
2. Join them to the ‘nrc’ lexicon using ‘right\_join’ and count the frequency of each sentiment occurring across the reviews.
3. Join them to the ‘bing’ lexicon using ‘inner\_join’ and count the frequency of each sentiment occurring across the reviews.

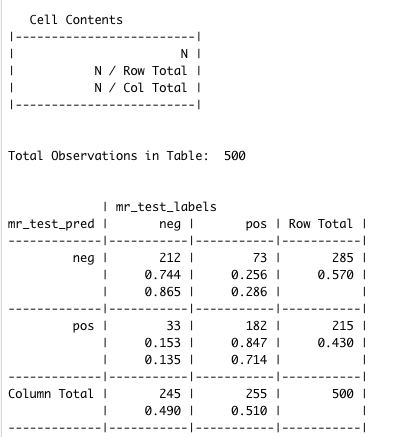
PERFORMANCE IMPROVEMENT

1. Compare the positive and negative sentiments and generate plots to gauge the overall emotions in the movie reviews.

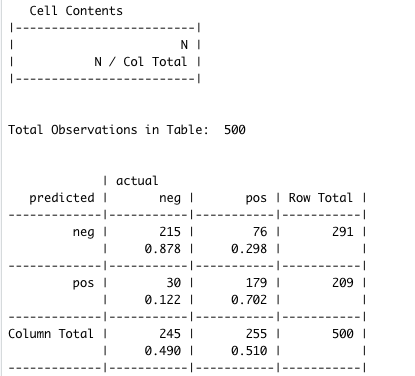
**RESULTS**

**NAÏVE BAYES:**

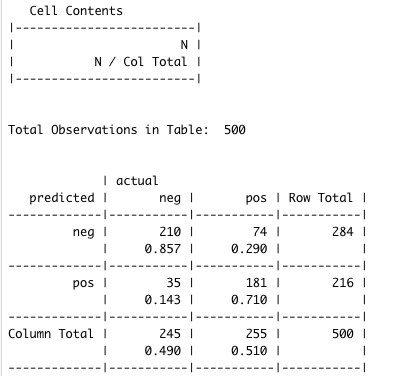
Cross Table for predicted versus actual positive and negative labels:



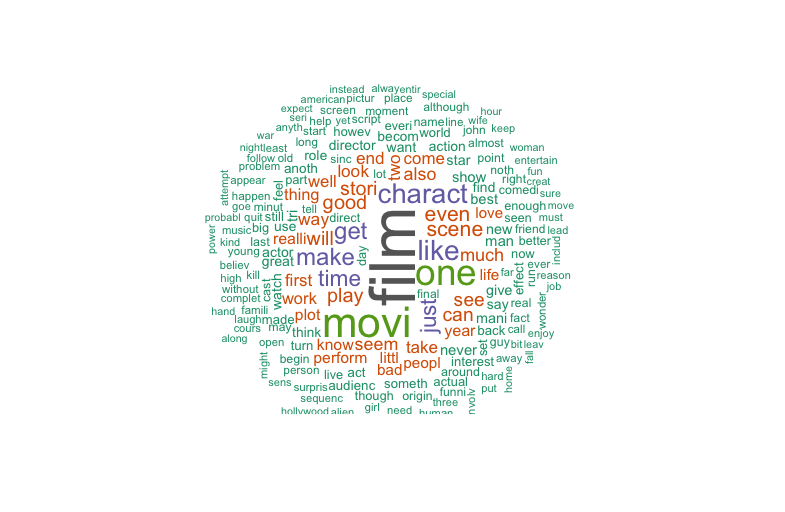
Cross Table for predicted and actual values after applying Laplace smoothing.



Cross Table for predicted and actual values after applying custom stop words and increasing frequency of words.



Word cloud of cleaned training data:

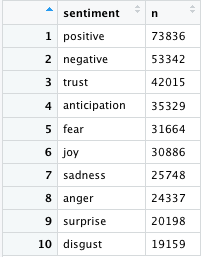


Word Cloud of positive and negative movie reviews respectively:



**RULE-BASED SENTIMENT ANALYSIS**

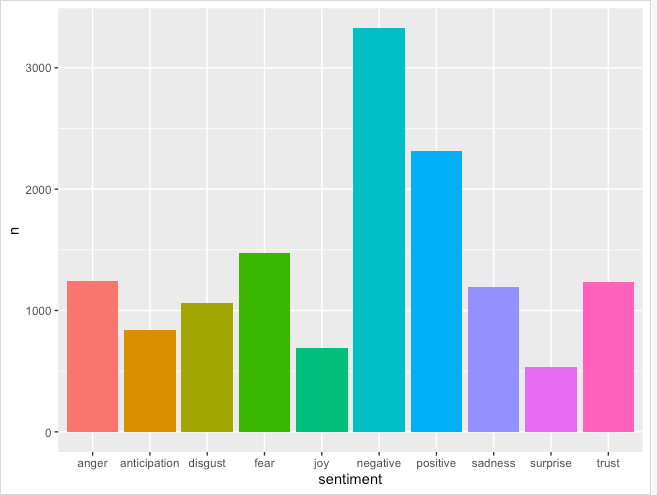
Word frequency reports for ‘nrc’ lexicon before the use of default stop\_words.



Word frequency reports for ‘nrc’ lexicon after the use of default stop\_words.

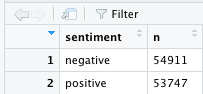


Frequency of sentiment words according to nrc lexicon classification.

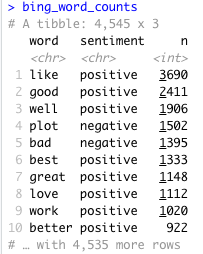


Word frequency reports for ‘bing’ lexicon:

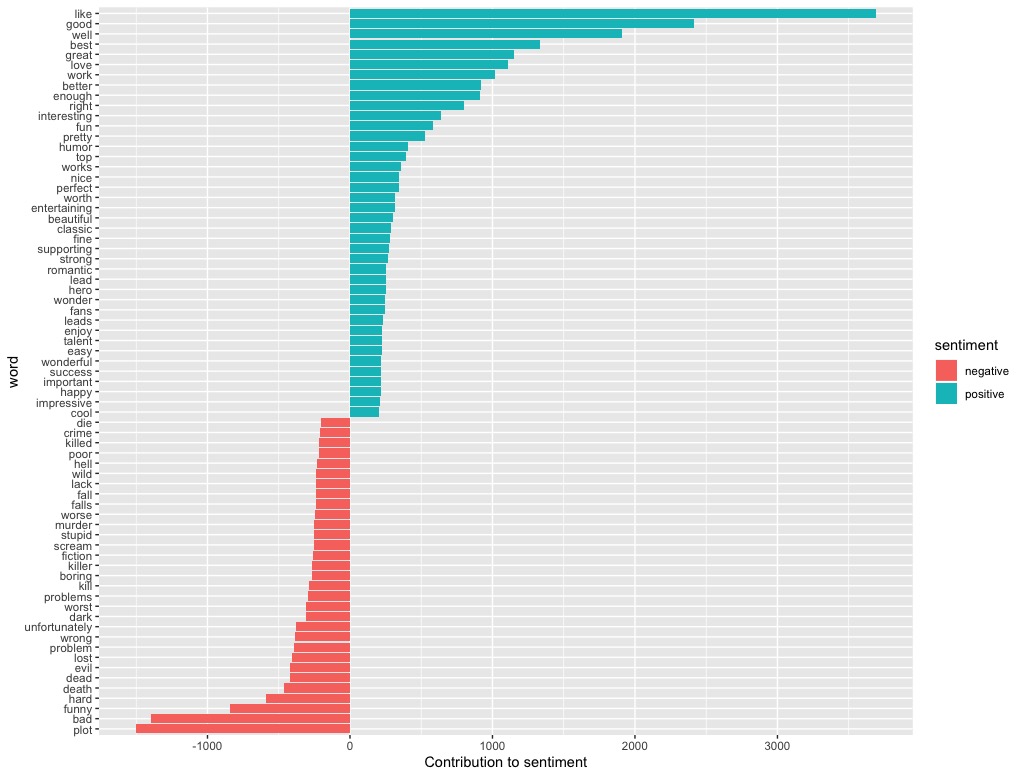
Binary classification of number of positive and negative words in the movie reviews.



Tibble representing the frequency of ‘bag-of-words’ present in the movie reviews.



Influence of the positive and negative emotions in the movie review



**REFERENCE**

Benoit Kenneth, Obeng Adam, Watanabe Kohei, Matsuo Akitaka, Nulty Paul, Müller Stefan (2020) *CRAN Repository: Package ‘readtext’ version 0.76* Retrieved from <https://cran.r-project.org/web/packages/readtext/readtext.pdf>

Boehmke Bradley (n.d.-a) *UC Business Analytics R Programming Guide: Text Mining: Sentiment Analysis* Retrieved from <http://uc-r.github.io/sentiment_analysis>

Boehmke Bradley (n.d.-b) *UC Business Analytics R Programming Guide: Text Mining: Creating Tidy Text* Retrieved from<http://uc-r.github.io/tidy_text>

Feinerer Ingo (n.d.) *R Documentation: Corpora* Retrieved from <https://www.rdocumentation.org/packages/tm/versions/0.7-7/topics/Corpus>

Gabriela De Queiroz, Colin Fay, Emil Hvitfeldt, Os Keyes, Kanishka Misra, Tim Mastny, Jeff Erickson, David Robinson, Julia Silge (2020) *tidytext: Text Mining using 'dplyr', 'ggplot2', and Other Tidy Tools* Retrieved from <https://cran.r-project.org/web/packages/tidytext/index.html>

Goyal Ankit, and Parulekar Amit (2015) *Sentiment Analysis for Movie Reviews* Retrieved from <https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf>

Grolemund Garrett, Wickham Hadley (2017) *R for Data Science* Retrieved from <https://r4ds.had.co.nz/tibbles.html>

Hu Xu, Bing Liu, Lei Shu and Philip S. Yu. (2004) *Opinion Mining, Sentiment Analysis, and Opinion Spam Detection* Retrieved from <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

Lantz Brett (2015) *Machine Learning with R,* 2nd ed, Livery Place: Packt Publishing Ltd.

Meyer David, and Xue Jinghao (n.d.) *Naïve Bayes Classifier* Retrieved from <http://math.furman.edu/~dcs/courses/math47/R/library/e1071/html/naiveBayes.html>

Mohammad Saif M. (2016) *National Research Council Canada: The Sentiment and Emotion Lexicons* Retrieved from <http://sentiment.nrc.ca/lexicons-for-research/>

MonkeyLearn (2020) *Sentiment Analysis* Retrieved from <https://monkeylearn.com/sentiment-analysis/>

Pang Bo, Lee Lillian (2004) *Movie Review Data: Sentiment Polarity Datasets* Retrieved from <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

R-core (n.d.) *R Documentation: cbind* Retrieved from <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/cbind>

Silge Julia (n.d.) *R Documentation: unnest\_tokens* Retrieved from <https://www.rdocumentation.org/packages/tidytext/versions/0.2.2/topics/unnest_tokens>

Silge Julia, and Robinson David (2020) *Introduction to tidytext* Retrieved from <https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html>

Wickham Hadley (n.d.) *Dplyr part of the tidyverse: Join two tbls together* Retrieved from <https://dplyr.tidyverse.org/reference/join.html>

Wikipedia (2020a) *Wikipedia: Sentiment Analysis* Retrieved from <https://en.wikipedia.org/wiki/Sentiment_analysis>

Wikipedia (2020b) *Wikipedia: Text Mining* Retrieved from <https://en.wikipedia.org/wiki/Text_mining>

Wikipedia (2020c) *Wikipedia Bayes Theorem* Retrieved from <https://en.wikipedia.org/wiki/Bayes%27_theorem#Statement_of_theorem>