Text Analysis of IMDb Movie Reviews

Akira Di Sandro, Sofia Fasullo, Amy Solano

2024-12-18

Contents

Introduction

Movie reviews inform curious film enthusiasts about how the work impacted individuals and allow people to decide whether or not they want to see the movie and reflect on movies they have watched in a new way. **IMDB** (or the Internet Movie Database) is one of the most famous and commonly used databases for movies and TV shows in the US. Users can do anything from exploring the release calendar, learning what movies and shows are ranked at the top of the nation in the current moment or the past, and finding reviews. These reviews can range from a paragraph to essays in length, and are written in several languages other than English.

In this study, we investigate the question, do shorter movie reviews contain more words with stronger sentiment compared to longer movie reviews? Are shorter reviews able to inform readers quickly and able to influence their opinions on a movie with stronger words to make up for the lack of elaboration that longer reviews are able to provide? On one hand, with less space to elaborate, one may think that shorter reviews may use more emotionally charged and pointed vocabulary. On the other hand, longer reviews have more space to use more words with strong sentiments. For our study, we use a dataset of 50,000 English IMDB movie reviews. We first clean our data using processes described below in the methods section below before performing sentiment analyses on each of the thirty shortest and longest reviews.

Methods

Data Preprocessing

Corpus

First, we want to convert our dataframe of movie reviews into a corpus. A text corpus (plural: *corpora*) "is a large and unstructured set of texts (nowadays usually electronically stored and processed) used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory." In this assignment, we utilize the r package `tm` for converting our data to a corpus and the subsequent preprocessing and analyses. Once we have created the corpus, we also make sure to convert everything to lowercase using `content_transformer` so that our analysis is not case-sensitive.

Punctuation and Special Characters

Since our corpus entries all consist of a single line, concatenation — the process of joining separate strings into a single line — is not necessary and thus we do not perform it, though this step may be important for preprocessing corpora that include line breaks. The next necessary step once we have created our corpus and converted all words to lowercase is to remove special characters, numbers, and punctuation marks.

To do so, we create functions `toSpace` and `remApostrophe` that take special characters (e.g., **@**, /,], $) as arguments, identifying instances of these characters in our corpus entries and replacing them with spaces. Other characters, like numbers and punctuation marks, can be removed using the `removeNumbers` and `removePunctuation` arguments within the `tm_map` function.

Stop Words and Word Grouping

After removing punctuation and special characters, we also make sure to get rid of **stop words**. In the English language, there are many stop words such as, "as", "and", "in", and "for", which are commonly used words that don't add useful information to the text overall. For this step, we used a predefined set of stop words from the `Snowball` package in R. We also removed several other neutral and common words in this corpus including words like, "movie", "feature", "screen", "imdb", and "film".

Stemming is a technique that helps us group words that have the same stem into the same word. For example, the process of stemming allows us to transform words like "singing", "singer", and "singers" to the stem word, "sing". In our analyses, we use the `stemDocument()` function from the `tm` package, which uses Porter's stemming algorithm, removing all suffixes to obtain the stem.

A shortcoming of stemming is the possibility that the ends of the words will be incorrectly or incompletely removed. In spite of this, this assignment favors stemming over other word-grouping techniques due to the simplicity of the operation for the purpose of analyzing movie reviews.

Similar to stemming, **lemmatization** also groups similar words together in a corpus. Unlike stemming, which groups multiple forms of a word together by removing the suffix, lemmatization searches for a word's *lemma*, or base or dictionary form, by conducting a more complete analysis of the words in the corpus. This makes lemmatization more computationally demanding than stemming, but depending on the analysis, the performance differences are modest. For this assignment, stemming is the better choice for grouping like words.

Word Clouds

A word cloud gives us an alternate way of examining a list of words and their frequency of occurrence in a corpus. It shows us the words that appeared the most in a corpus by giving the most frequently used words

a larger font size. Within the `wordcloud` function, we specify that only words that appear in **5,000 reviews of more** should appear in the word cloud, ensuring that the most relevant terms are displayed given that our corpus includes 50,000 reviews.

Sentiment Analysis

After cleaning our corpus and simplifying it to the best of our abilities to only include relevant words with meaning that could be analyzed, we perform a **sentiment analysis**. A sentiment analysis takes a corpus as an input and uses a lexicon to determine what the overall tone of a corpus is. Sentiment analysis looks different depending on what lexicon the researcher decides to use.

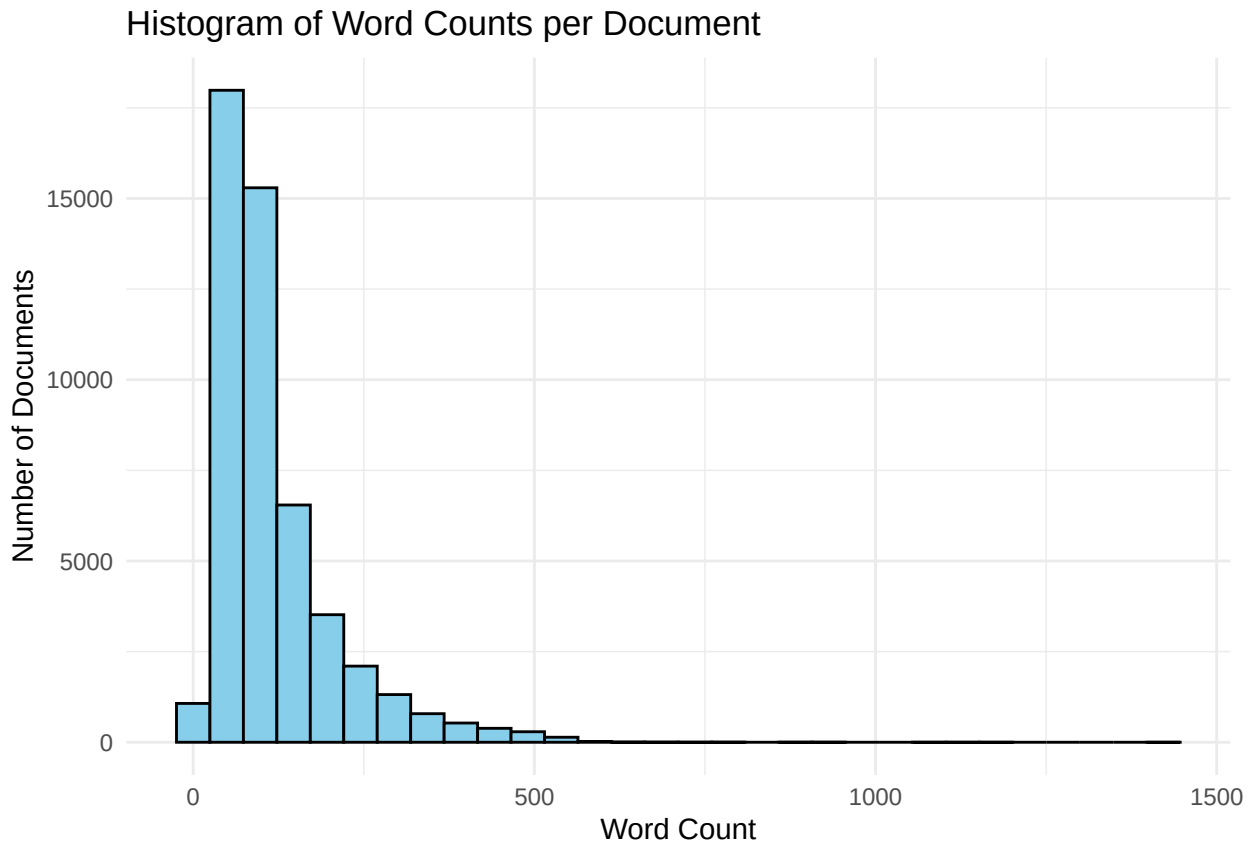
A **lexicon** is a list or dictionary of words or phrases that is coded with a sentiment or emotion. Some lexicons such as the National Research Council (NRC) lexicon (also known as the EmoLex) provide a dictionary of words and their association to emotions (anger, joy, sadness, fear, disgust, anticipation, trust, and surprise) in addition to sentiment (positive or negative). Other lexicons use numeric values to code the intensity of the sentiment. Our study utilizes one such lexicon, the **AFINN lexicon** available in the `syuzhet` package. In the AFINN lexicon, each word is assigned a score ranging from negative (-5) to positive (5). Utilizing this lexicon allows us to compare sentiment across corpus entries of different lengths to answer the research question of whether shorter movie reviews contain more words with stronger sentiment compared to longer movie reviews.

There are a couple limitations to sentiment analysis that are important to highlight. One such limitation is that positive and negative words may wash each other out when performing sentiment analysis on a big body of text, resulting in neutral overall sentiment. For this reason, sentiment analysis works best on smaller chunks of text since they are more likely to have a consistent sentiment throughout. In our analysis, we explore sentiments of movie reviews, which may have some mixed sentiments, but we assume that our reviews are short enough that the sentiment being communicated in each review is consistent. Additionally, sentiment analysis is not able to decipher things like sarcasm and negative words preceding words (that would flip the sentiment of the text).

To explore our research question, we take the 30 shortest and 30 longest reviews, after data cleaning. For each review, the sentiment score is calculated by taking the frequency of each word, multiplying it by the AFINN lexicon scores, and summing together all the word scores in a review. To account for the different lengths of the corpus entries, an average score is also calculated by dividing the sentiment score by the total number of words in the review. Calculating the average allows us to more meaningfully compare the reviews and see the intensity of the sentiment.

Results

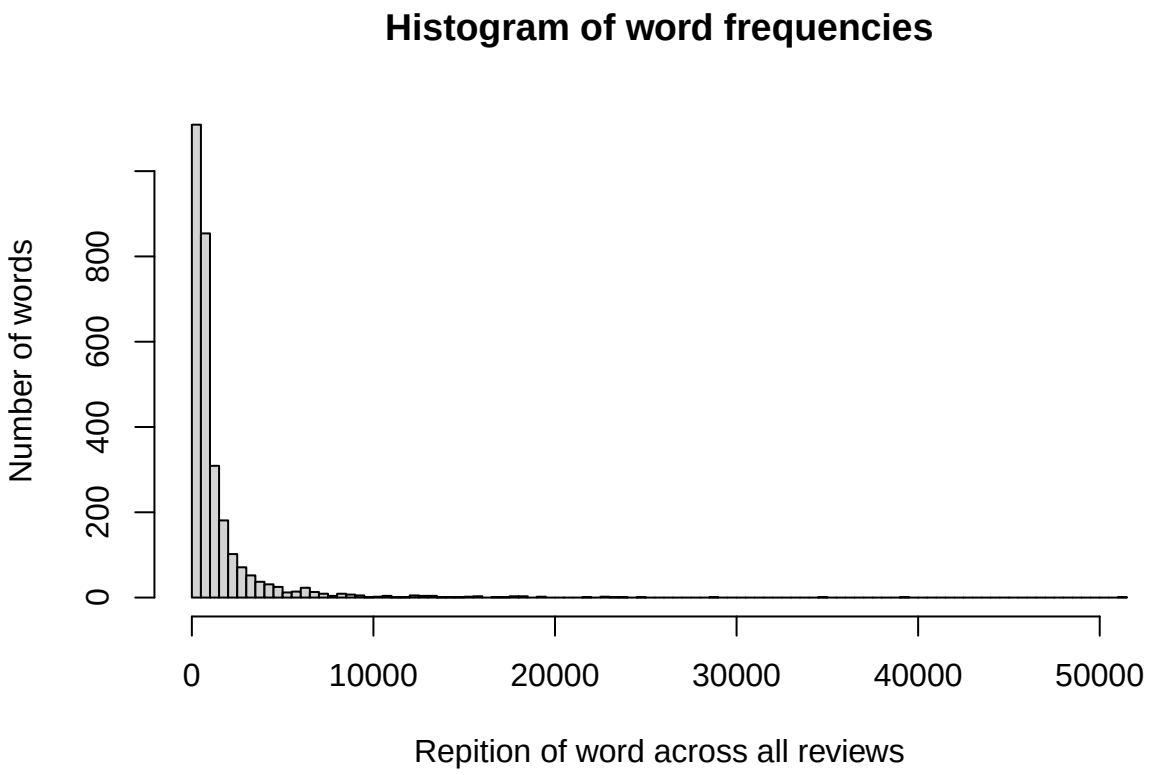
Data cleaning



The above histogram shows the range of review lengths in our corpus after data cleaning. A large portion of the reviews fall between 0 and 250 words. Below, we see the actual range of our review lengths.

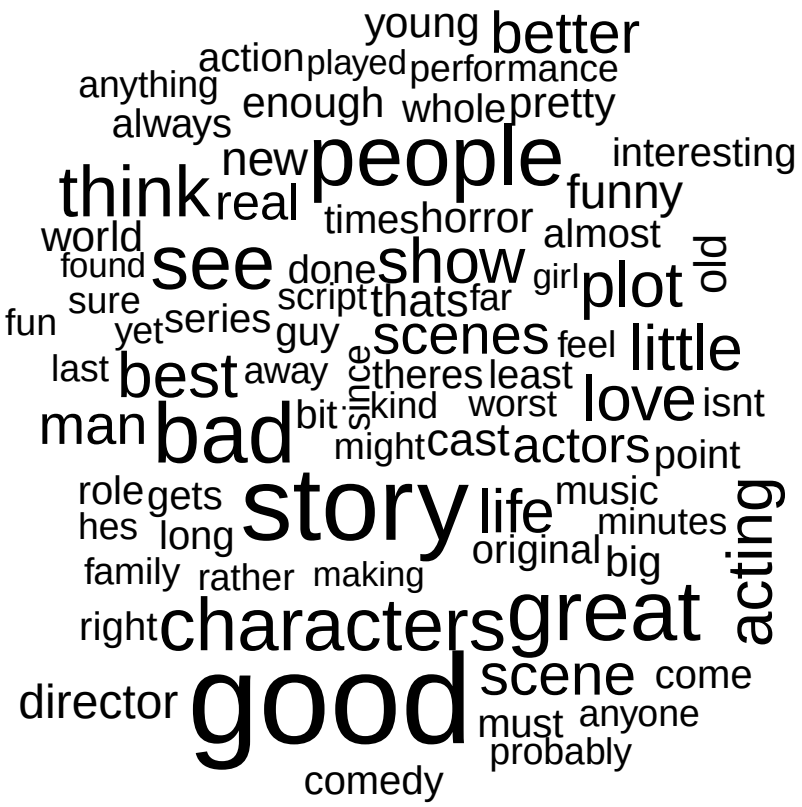
```
## [1] "The shortest reviews had 3 words and the longest reviews had 1425 words."
```

Word Cloud



Ideally, the word cloud should contain around 200 of the most commonly used words. From the histogram of word frequencies, it appears most words (terms) were repeated between 1-100x per review. There are some outliers as the histogram tapers to the right, so it may be that there are still filler words that need to be removed.

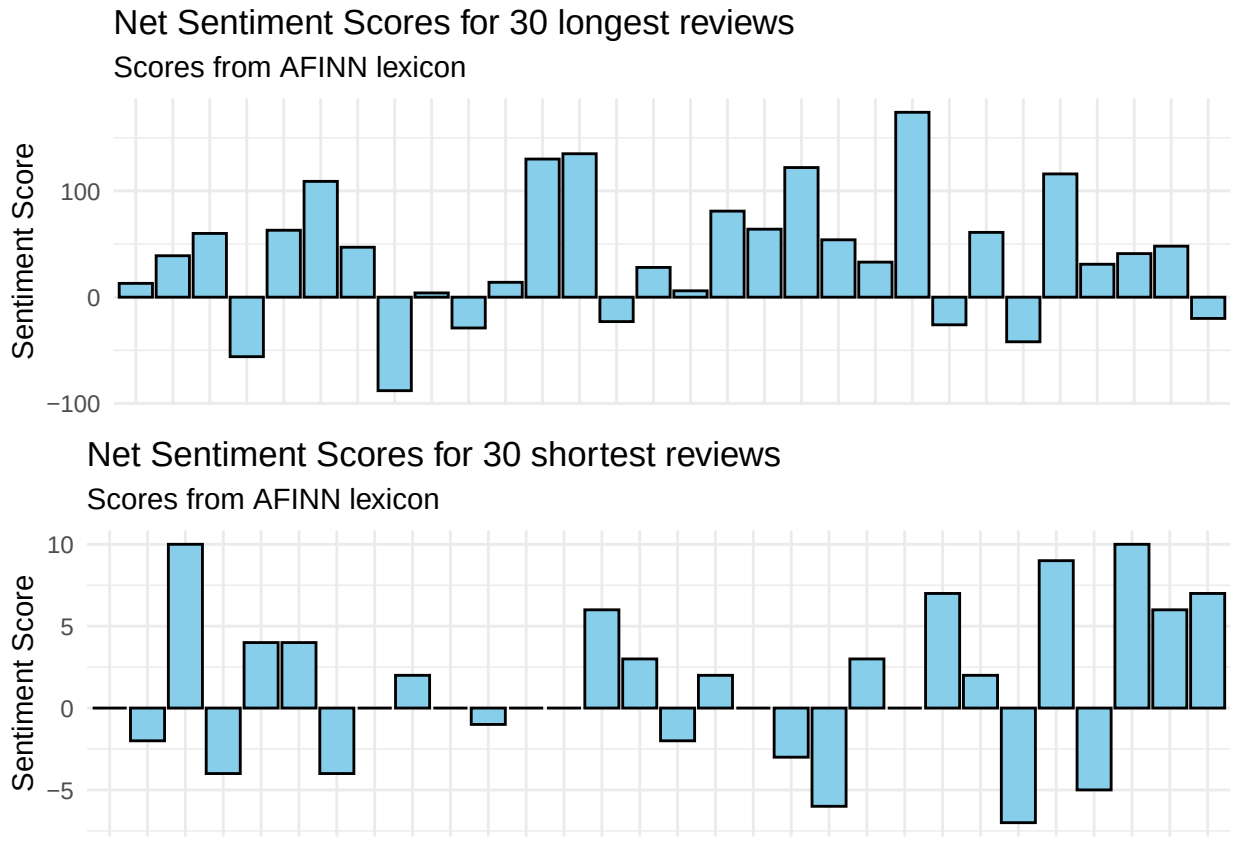
```
myCorpus <- tm_map(myCorpus, removeWords, c("one", "like", "just",
    "even", "time", "really", " see", "can", "much", "well",
    "get", "will", "also", "first", "dont", "movies", "made",
    "make", "films", "way", "watch", "many", "seen", "two", "character",
    "never", "know", "ever", "still", "say", "end", "something",
    "back", "watching", "thing", "doesnt", "now", "didnt", "years",
    "another", "though", "actually", "makes", "nothing", "find",
    "look", "going", "work", "lot", "every", "part", "cant",
    "want", "quite", "things", "seems", "around", "got", "take",
    "however", "fact", "give", "thought", "ive", "may", "without",
    "saw"))
```



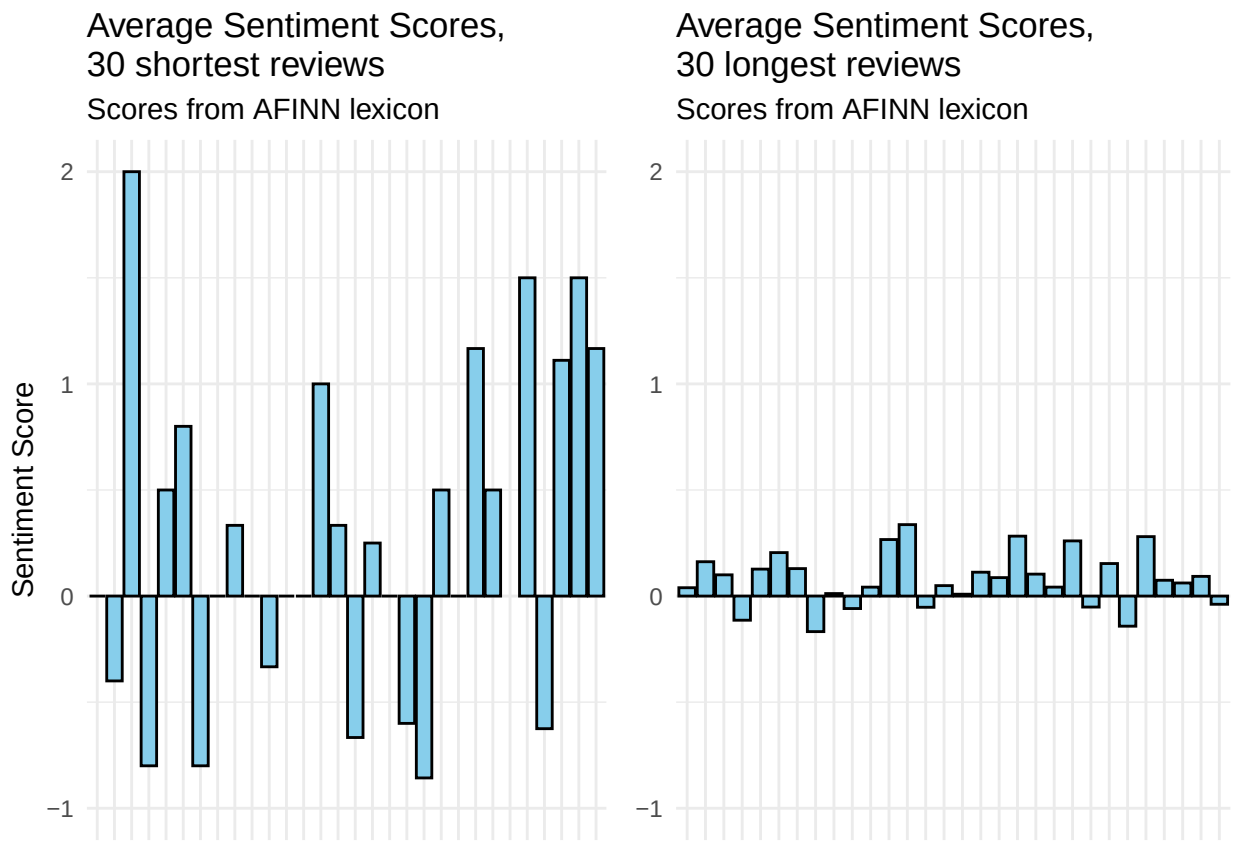
After clearing out additional filler terms, the word cloud highlights terms like "good", "bad", and "characters." There are many terms that have to do with production, like "cast", "director", "script", and "acting." This suggests that reviewers are not solely focused on the plot, but on the behind-the-scenes as well. Perhaps the focus is on a certain actor or director that influences the reviewer's watching decisions. These words on their own do not suggest a purely positive or negative reaction, and there seem to be few words that are negative on the surface, like "worst."

In the word cloud, we also observe some of the negative outcomes of stemming with the word "scene" and "scenes" both appearing, despite having the same stem. The stemming function failed to properly split these words.

Sentiment Analysis



The above plots show the results of our sentiment analysis as the net score for each review in the 30 longest and 30 shortest reviews. The combined scores for the shortest reviews only go as high as ten, which makes sense given the low word counts. The scores for the longest reviews still have low scores, but the range is wider on both the positive and negative end. The following graphs look at the average score, the sentiment divided by the word count, side-by-side on the same scale.



Looking at the averages, the shorter reviews have a greater range on both the positive and negative ends compared to the longest reviews. All of the scores for the longest reviews fall between -0.5 and 0.5, while the shortest reviews range from -1 to 2. This suggests that the shorter reviews may convey sentiment more effectively, though it may also be true that the shorter length may play a part in the score due to the process of calculating averages.

Discussion

Our corpus as a whole contains long and short reviews with a range of sentiment scores. Our word cloud suggests that the production of a film or show is very important in the review process. Our sentiment analysis suggests that longer reviews use fewer words of more intense sentiment and thus do not lean strongly towards positive or negative. One thing our analysis did not consider was the existence of mixed reviews, which are quite common in the entertainment industry. In the future, we could use different lexicons to more effectively capture mixed and neutral reviews, looking more at frequency of a certain type of review in addition to the scores used in this review. We could also use more reviews to help us answer our research question, using a word count threshold rather than cutting off at a certain number. While our sentiment analysis did answer our research question, our word cloud and the consideration of mixed reviews leaves room for further exploration of our research question, using different methods that could capture more nuance in the reviews.

References

1. Hill, Chelsey. 2023. “Sentiment Analysis (Lexicons)”. Rstudio-Pubs-Static.S3.Amazonaws.Com. https://rstudio-pubs-static.s3.amazonaws.com/676279_2fa8c2a7a3da4e7089e24442758e9d1b.html.

2. “Sentiment Analysis In R | R-Bloggers”. 2021. R-Bloggers. https://www.r-bloggers.com/2021/05/sentiment-analysis-in-r-3/.

3. Robinson, Julia. 2023. “2 Sentiment Analysis With Tidy Data | Text Mining With R”. Tidytextmining.Com. https://www.tidytextmining.com/sentiment.html.

4. “Text Mining: Sentiment Analysis · AFIT Data Science Lab R Programming Guide”. 2023. Afit-R.Github.Io. https://afit-r.github.io/sentiment_analysis.

5. “TDM (Term Document Matrix) And DTM (Document Term Matrix)”. 2023. Medium. https://medium.com/analytics-vidhya/tdm-term-document-matrix-and-dtm-document-term-matrix-8b07c58957e2.

6. “Text Clustering With R: An Introduction For Data Scientists”. 2018. Medium. https://medium.com/@SAPCAI/text-clustering-with-r-an-introduction-for-data-scientists-c406e7454e76.

7. “Introductory Tutorial To Text Clustering With R”. 2023. Rstudio-Pubs-Static.S3.Amazonaws.Com. https://rstudio-pubs-static.s3.amazonaws.com/445820_c6663e5a79874afdae826669a9499413.html.

8. “Library Guides: Text Mining & Text Analysis: Language Corpora”. 2023. Guides.Library.Uq.Edu.Au. https://guides.library.uq.edu.au/research-techniques/text-mining-analysis/language-corpora.

9. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, “Introduction to Information Retrieval”, Cambridge University Press. 2008. https://www-nlp.stanford.edu/IR-book/