# Sean McLean

# ALY 6040

# Professor Reilly

# Module 5

# Technique Practice

## Introduction

The technique project in module 5 utilizes text extracted from "The Hobbit" novel as a practical exercise in data mining, employing text mining and analysis methods. The objective is to preprocess the data for natural language processing and identify the most commonly occurring words in the book. Analyzing these words offers insights into the narrative's themes and essence, derived from their high frequency. This process is essential in text mining as it transforms unstructured data into structured data, facilitating further analysis and interpretation.

## Code Walk Through

The packages ‘tm,’ ‘SnowballC,’ ‘wordcloud,’ and ‘RColorBrewer’ are installed and loaded for text mining, text stemming, word-cloud generating, and the color palettes, respectively. After installing and running the necessary libraries for the project, the ‘filePath’ function is implemented to upload The Hobbit text that will be used for the text mining process. A text variable is then created by using the ‘readLines’ function on the ‘filePath’ text that will open the connection of the file and read the lines that are contained. The ‘warn’ argument is added as a false equation because it does not contain any null values. The data is then loaded into a new variable called ‘docs’ by using the ‘corpus’ function and by putting the text into a vector using the ‘VectorSource’ for interpretation. The text from the ‘docs’ variable is then presented and analyzed by using the ‘inspect’ function.

The editing process of The Hobbit text is then executed to clean up any unnecessary data and make the text more presentable after the revisions. The first step is to jettison the symbols by creating a new variable called ‘toSpace’ and then using the ‘content_transformer’ function to define a custom transformation for modifying the content of each document in a corpus. The ‘toSpace function,’ created with ‘content_transformer,’ replaces specified patterns in the text with spaces. Subsequently, the ‘tm_map’ function is employed three times to apply these transformations to each document in the ‘docs’ corpus. The initial ‘tm_map’ call replaces all instances of ‘/’ with spaces, while the subsequent two calls replace ‘@’ and ‘|’ occurrences with spaces, respectively, in every document. This likely serves as a preprocessing stage to ready the text data for further analysis or modeling.

The next steps in the editing process entails the execution of a sequence of text preprocessing operations utilizing the ‘tm_map’ function within the R package ‘tm.’ Initially, it standardizes the text by converting all characters to lowercase. Following this, it eliminates numerical digits, as well as common English stop words such as "the" and "and", and custom stop words like "said" and "like". It also removes punctuation marks and any redundant white spaces in the data. These preprocessing steps are commonly applied to refine and normalize text data before further analytical or modeling procedures. After these revisions have been made, the text is again evaluated using the ‘inspect’ function where the text is noticeably edited from the initial import of the data.

The text is then combed through to find the most familiar words and the frequency of each word in the file. This task is completed by creating four different variables that will show the top 10 words and how often they appear. The new ‘dtm’ variable turns the ‘docs’ variable into a term document matrix and then uses the ‘as.matrix’ function to turn the ‘dtm’ variable into a new variable ‘m’ with a matrix for the set of values. The data from the ‘m’ variable is then created into a new variable called ‘v’ and is sorted in descending order for the numeric values of each word using the ‘sort’ function and the ‘rowSums’ function. The data frame with these values is then created into a new variable called ‘d’ with new column names of ‘names’ and ‘freq.’ The head of the data frame is executed using the ‘head’ function showing the top 10 names by frequency.

After the ‘set.seed’ function is run to generate random numbers at value ‘1234’, the ‘wordcloud’ function is plotted to show the words and frequencies, with the minimum frequency set to one and the maximum frequency set to 200. Other arguments used are ‘random.order’ function that shows random words in decreasing frequency if false, the ‘rot.per’ function that sets the proportion of word with 90-degree rotation, and the ‘colors’ function for frequency purposes (See Appendix A). The ‘findFreqTerms’ function is then utilized to find the most frequent items in the file with the ‘lowfreq’ argument for the lower frequency bound set to 100. The ‘findAssocs’ function is then run showing the association in percentages between the word “gold” and all the other words in the file. The ‘corlimit’ argument is also executed with the lower correlation limit set to 0.1 to just show associations above that value. A bar plot is then created to show the top 10 words from the ‘d’ variable in terms of their frequencies and are arranged in descending order (See Appendix B).

## Analysis

The alleviation of unnecessary words, grammar and punctuation, and symbols in the preprocessing phase makes the text file a jumble of meaningful words for text analysis. This is a valuable step in the natural language processing tasks in that it removes stop words and gets the remaining text into a structured format. The top 10 words by frequency are all types of nouns with ‘Bilbo’ being by far the most common word in the text file at 503 occurrences. The words ‘dwarves’ and ‘Thorin’ which are also characters in the book are also among the top 10 most frequently used words. The remaining seven words (now, one, came, long, back, time, come) are common nouns that describe things (Scribbr, 2024). The generated word cloud confirms the validity of the 'dtm' variable established to identify the most prevalent words, prominently displaying 'Bilbo' as the overwhelmingly popular term in large, bold letters against a dark background. The other most familiar words that are in the top 10 are provided in large pink letters as displayed in appendix A below.

The ‘findFreqTerms’ function displays a list of around 50 frequent words used in the book that are a mix of nouns, verbs, and adjectives. All 10 of the most frequent words detected previously are included in this list. The ‘findAssocs’ function executes the most associated words with the

selected word ‘gold’ with the words ‘silver’ and ‘jewels’ having the highest associations values at 0.21. There are about 50 words or so in the book that are above the set limit in the code at 0.1. The bar chart that shows the top 10 most frequent words is a strong visualization with the color selected and the words are in descending order.

## Interpretations & Recommendations

An evaluation of those words seems to indicate that the plot could pertain to some sort of expedition or quest where time is imperative, and the distance of the travels is far. The ‘findFreqTerms’ function seems to be an extension of the top 10 most popular words with a lot of words describing characters or places. It also contains words that relate to a potential expedition in the books and uses words that elaborate on it. Some of the words that could relate to this quest are ‘went,’ ‘left,’ and ‘away.’ The ‘findAssocs’ function contains words that are related to the word ‘gold’ in many ways that can be looked at from an angle of named entity recognition. Some of them are monetary like the words ‘costly,’ ‘values,’ and ‘worth,’ and some of them pertain to potentially the location of the gold, like ‘pots,’ ‘rivers,’ and ‘fountain.’ Other words in the mix are verbs like ‘gleamed,’ thriceforged,’ and ‘hoarded’ that could describe what kind of actions are being taken with the gold (Cambridge Dictionary, 2024). These relationships can help formulate potentially what the expedition could be about the characters and the usage of gold to explain the plot of the book.

For an individual that has not read the novel this execution of text mining and key phrase extraction can be utterly useful in describing a book and creating a summary of what the book is about. The frequency of words could also be used in recommendation system building for platforms like Audible for using key words or phrases for recommending books to subscribers. I recommend word tagging the most frequent words to study the associations between them that show this potential plot. I would also modify the sizes of the matrix, wordcloud, ‘findFreqTerms,’ and ‘findAssocs’ functions to identify more words and relationships between words that could provide more input into the storyline. Some added visual methods like syntactic parsing can also provide a cleaner breakdown of speech tagging. I recommend these steps that could lead to an accurate description of the Hobbit and the book's plot.

## Conclusion

Exploring "The Hobbit" text through data mining has offered both hands-on experience with text mining techniques and valuable insights into its central themes. By scrutinizing the most frequently-used words, we have uncovered deeper layers of the story's essence and key components. This exercise underscores the importance of text mining in transforming raw data into organized information, thereby empowering us to derive significant insights from textual data. Looking ahead, the skills and insights gained from this project will undoubtedly serve as invaluable assets in future data mining projects.

## References

Cambridge Dictionary. (2024). *Major word classes.* Cambridge Dictionary. https://dictionary.cambridge.org/us/grammar/british-grammar/word-classes-and-phrase-classes

Scribbr. (2024). *What Is a noun? | Definition, Types & Examples.* Scibbr.com. https://www.scribbr.com/category/nouns-and-pronouns/

# Appendix

## Appendix A


## Appendix B