MA331 COURSEWORK ON SENTIMENTAL ANALYSIS

Registration Number: 2010655

31/03/2021

1.Introduction

The main objective of this paper is to carry out a sentimental analysis of two books from the Project Gutenberg collection. Project Gutenberg is a public library of more than 60,000 custom e-books. Sentiment analysis is an effective way to explore an author's feelings towards a subject. This paper aims to analyze two books, "Alice's Adventures in Wonderland" by Lewis Carroll from child list and "Great Expectations" by Charles Dickens from the adult list from the Gutenberg Collection, and compare the overall emotions and sentiments throughout these works using String processing and Text Mining.

"Alice's Adventures in Wonderland" is a story about a little girl Alice who falls down a rabbit hole and delves into a fantasy world filled with wonderful people and animals full of surprises whereas "Great Expectations" tells us the story of an orphan boy who is adopted by a blacksmith's family, who has good luck and great expectations, and then loses both his luck and his expectations. However, he learns the importance of friendship and love and, becomes a better person for it.

2.Method

2.1 Preliminary Data Analysis

To start the data analysis, I have assigned the dataset "Alice's Adventures in Wonderland" to a variable called "child" and the dataset "Great Expectations" to a variable called "adult". The child dataset consists of 3380 rows and 2 columns while the adult dataset consists 20024 rows and 2 columns. For the convenience of text mining, next step is converting the data frames to Tibbles. The benefits of Tibbles compared to data frames are Tibbles has a nice printing method that displays only the first 10 rows and all the columns that fit on the screen. This is convenient when you work with large data sets. When printed, the data type of each column is specified.

2.2 Tidytext format

The first step in the sentimental analysis is to convert the data sources (here the 2 books) into a tidy text format which will allow us to carry out the analysis easier. The structure of the tidytext format is defined as each variable is treated as a column, each observation is treated as a row and each observational unit is considered as a table.

To get the dataset in a tidytext format we need to restructure it into a one-token-per-column format. A token can be a word, character, subword, etc. Tokenization is a process of separating a piece of text into smaller units called tokens. In most cases, tokenization is done based on white space that is splitting text by words (i.e. the default)

To perform tokenization and transform the data into a tidy data structure that is one token per column format we can use the tidytext's unnest_token() function. This is a useful function which not only performs the tokenization but also performs operations such as removing punctuations and converting the tokens to lowercase so that we can easily compare or merge with other datasets.

2.2.1 Removing stop words

The next step is removing the stop words in the text. When analyzing the context of the text it can be seen that the dataset contains so many stop words such as "the", "as", "and", "of", "to" etc. these words are useless for our sentimental analysis so we can remove these words using tidytext anti_join() function.

2.2.2 Counting the most common words

To count the most common words in the books we can use dplyr's count() function

2.3 The sentiment datasets

There are a diversity of methods and dictionaries that are available for assessing the view or emotion in text. The tidytext package offers access to numerous sentiment lexicons. Three general-purpose lexicons are as follows

a) Afinn from Finn Årup Nielsen: The AFINN lexicon allocates words with a score that turns between -5 and 5, with negative scores representing negative sentiment and positive scores representing positive sentiment.

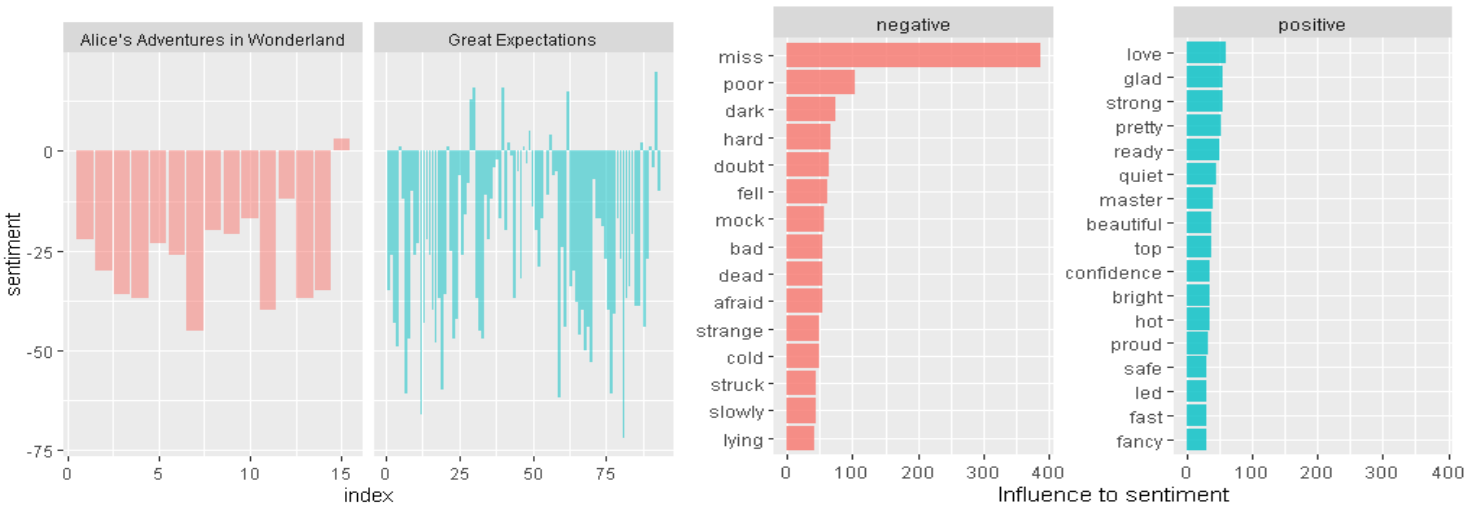
b) Bing from Bing Liu and collaborators: This lexicon classifies words in a binary fashion into positive and negative categories.
c) Nrc from Saif Mohammad and Peter Turney: This lexicon classifies words in a binary manner ("yes"/"no") of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.
All of these three lexicons are grounded on unigrams (single words). These lexicons comprise numerous English words that describe emotions like joy, anger, sadness, and so on, and each word has a specific score based on positive or negative sentiment.
Now, let's have a look at the sentiment dataset to measure the different sentiments that are embodied across the books.

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
## # ... with 6,779 more rows
```

The above table gives an overall idea about the sentiment words in the sentiment dataset. Now we aim to analyze the sentiments separately among each chapter of each book as the story progresses. To achieve this creating an index that splits the 2 books by 600 words, which is an approximate count of words on every two pages so it will allow us to evaluate deviations in sentiments even within chapters. Then inputting "bing" lexicon to inner_join() function to access the positive vs. negative sentiment of each word. Then counting the positive and negative words in every two pages. We then use spread() so that we get negative and positive sentiment in separate columns, and finally calculate a net sentiment (positive - negative) and plotting with the help of ggplot.

3.Results

3.1 Comparing the two books using Bing Lexicon

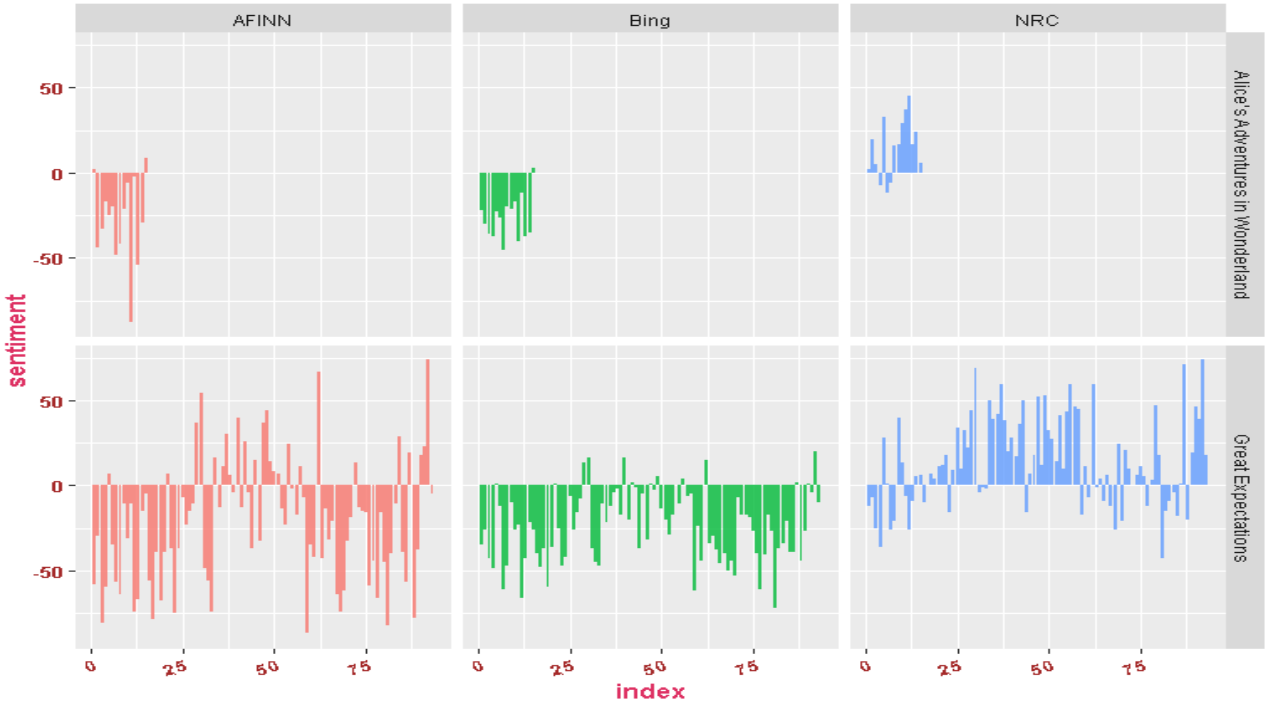


The plot shows how each book fluctuates towards more positive or negative sentiment throughout the story.
The first plot compares the sentiments in each book as the story progresses based on bing lexicon. It has been clear that both books have more shift towards negative sentiment. While the second plot shows the contribution of each word towards positive or negative sentiment as per bing lexicon.

3.2 Comparing Sentiments based on each lexicons

Through numerous options for sentiment lexicons, we might need some more information on which one is proper for our purposes. Let's analyze all three sentiment lexicons and observe how the sentiment changes across the books. Afinn lexicon estimates sentiment with a numeric score between -5 and 5, while the other two lexicons classify words in a binary fashion, either positive or negative. So we need to create a separate pattern for Afinn and another one for bing and nrc.

The three different lexicons for conniving sentiment bounce results are diverse in an absolute sense but have similar relative trajectories through the books. It has been noted that the afinn and bing lexicons are shifted to gives the largest absolute values, with high negative values throughout the books while the nrc results are shifted higher relative to the other two, labeling the text more positively, but detects similar relative changes in the text. Now, let's look into the variations of these lexicons separately in each book.



From this it is clear that the lexicons afinn and bing shows a negative shift throughout the book “Alice’s Adventures in Wonderland” whereas the nrc lexigons shifted more positive. But for “Great Expectations” there is a huge difference in shifts for all the 3 lexigons.The ratio of positive to negative words in afinn is high compared to other two and is shifted more to negative. Bing also shifted more to negative whereas the nrc lexicon is shifted to more positive.

Lets create 2 two tables in which one contain the list of negative words and the other contain list positive words from the Bing lexicon which are present in each book and calculate percentage of negative/positive words per each chapter. The table only shows the top one chapter in each book contain more percentage of positive or negative words.

```
## # A tibble: 2 x 5
##   Book                       Chapter Negative_words  words percentage
##   <fct>                        <int>          <int>  <int>      <dbl>
## 1 Alice’s Adventures in Wonderland     6             85    789       10.8
## 2 Great Expectations                  44            116    777       14.9
```

```
## # A tibble: 2 x 5
##   Book                          Chapter Positive_words words percentage
##   <fct>                           <int>          <int>  <int>      <dbl>
## 1 Alice's Adventures in Wonderland     1             32    616       5.19
## 2 Great Expectations                  17             86    885       9.72
```

From the above tables, we can see that chapter 6 of "Alice's Adventures in Wonderland" has more negative words and chapter 1 has more positive words whereas for "Great Expectations" chapter 44 contains more negative words and chapter 17 contains more positive words according to bing lexicon.

3.3 Wordclouds

Alice's Adventures in Wonderland

Great Expectation

Overall positive and negative words within 2 books

4. Discussion

Sentiment analysis offers a method to realize the attitudes and feelings spoken in texts. we explored how to tactic sentiment analysis employing tidy data principles. We can effectively convey an Author's emotions in text using lexicons but the main limitation of this approach is different lexicons show some kind of variations for the same text. For example, if some words are treated as negative in one lexicon the same word may be treated as positive by other lexicons. But overall it shows the same shifts throughout the trajectory of the book. The main challenge is to choose the right lexicon which matches our purpose.

5. References

Google: (https://www.tidytextmining.com/sentiment.html)

Kaggle: (https://www.kaggle.com/rtatman/tutorial-sentiment-analysis-in-r)

GitHub: (https://github.com/bradleyboehmke/Text-Mining-Tutorials/blob/master/02-sentiment.md)