Project Overview

For this project, the data source we choose is Project Gutenberg. We selected two books, *Camille* and *The Adventures of Tom Sawyer*, for the detailed comparison, and five more books[1] for showing the clustering. We analyzed the source going through three steps: characterizing word frequencies, computing summary statistics[2], and doing natural language processing using sentiment analysis, text similarity, and clustering. Through the project, we hope to learn how to directly harvest data from online sources, store and clean up the data, and conduct the analysis using python.

Implementation

At the beginning stage, we focused on two texts from Project Gutenberg. We implemented the project first by inputting the texts from url into string (text1 and text 2). We skipped the header and the end of the texts, excluded the punctuation and whitespace in the texts and computed a hist that can show a map from each word to the number of times it appears. With the function most_common and also excluding the some stopwords, we checked for the top ten most frequent words in both books. However, when we ran the code to compute the most common words, we realized that some common words showed up as names of the characters, which are meaningless to analyze for the texts. Therefore, we decided to exclude the names in the top ten most frequent words (marguerite, tom, and huck). In addition, we stored the top 100 most common words into dictionaries, and used the function "subtract" to find the differences between two books' top 100 most common words.

After conducting the basic text analysis, we applied natural language processing by first using sentiment analysis to figure out whether the reviewer has a negative (when the compound value <= -0.05), positive (when the compound value >= 0.05), or neutral feeling toward the movie through their words. Additionally, we analyzed the similarities of two texts using the cosine similarity value. When we finished comparing these two texts, we realized that the larger

[1] *Alice's Adventures in Wonderland* by Lewis Carroll*, Adventures of Huckleberry Finn* by Mark Twain*, The Tragedy of Pudd'nhead Wilson* by Mark Twain*, The Prince and The Pauper* by Mark Twain*, Through the Looking-Glass* by Lewis Carroll

[2] Reflect the top 10 most frequent words and export the 100 most common words in one book but are not in the other one

sample size is, the more significant our clustering will be. It also offers us a better understanding and interpretation of the similarities between different books. For that reason, we selected five more books for the clustering- 2 of them were written by Lewis Carroll and 3 of them were written by Mark Twain. And the result may help us to see various styles of different authors.

Throughout the implementation, when we tried to compare the top 100 most common words in two books, we debated whether we should store the words in dictionaries or in lists. Eventually, we decided to store them in the dictionaries since it would be easier for us to store the words in the dictionaries while checking the frequencies in the former step. And we employed the function "subtract" from our homework, which could be used directly for comparing and computing the different items between two dictionaries.

Results

Based on our text analysis of the books, we found that some of the most frequently used words in both books are "said", "one", and "see". (See the result from **Appendix 1**). It is interesting as we saw the output of the histograms since we could speculate that these words might also be some of the most commonly-used words in English literature, especially those written in the 19th century (the two books we choose are both from this period). According to Wikipedia, "100 Most common words in English"[3], all three of them are on the list. Thus, it proves that people nowadays in real life also use these words at a high frequency. In addition, by comparing the two texts, the program generates 100 most common words in one book but are not in the other one (See the result from **Appendix 2**). By looking at the output, we could get a rough image of what the book is about without reading it. For example, when looking at the words listed in the 100 most common words in *Camille* that aren't in the 100 most common words in *The Adventures of Tom Sawyer*, we get the idea that this novel is probably about romance and relationships. Vise versa, the theme of the other book is about children and adventure. Hence, this program helps people who never read particular books to comprehend the basic themes of the novels.

[3] "Most common words in English - Wikipedia." https://en.wikipedia.org/wiki/Most_common_words_in_English. Accessed 20 Oct. 2020.

Based on the output of the Natural Language Toolkit, the overall sentiment of the sentences is rated as negative in *The Adventures of Tom Sawyer*, while for *Camille,* the sentences are overall rated as positive (See the result from **Appendix 3**). It is quite surprising due to the nature of these two books. In *Camille*, Marguerite, the protagonist, experiences a miserable life and eventually dies because of tuberculosis. The book presents the tragedy of the camellia from blooming to withering. Thus, it came into our notice that the result did not fall into our expectation. Perhaps the author, Alexandre Dumas fils, attempts to employ the romantic writing style for the story plot in order to form a stronger contrast with its tragic ending. And according to our reading, the author puts intensive emphasis on the luxury life of the upper class in Paris. Hence, we concluded that it made sense for the NLTK program to print out a positive outcome. As for *The Adventures of Tom Sawyer*, the overall story plot shows a joyful atmosphere that illustrates a boy's adventure. Just like many other adventure stories, the protagonists defeat the villain and find the treasure. Yet the author, Mark Twain, applies his own childhood memories to the story. Under the cover of "an adventure story for children", Twain unveils the reality of the American society in the 19th century and criticizes the children's education of the middle class. He also emphasizes greed and hypocrisy in morality through the character depiction. After interpreting the book, we came back to the conclusion of negative sentiments. And so it makes more sense now for the program to have such output.

Moreover, by clustering words from different texts written by the same author, we thought we are able to speculate about the unique writing style of an author (See the result from **Appendix 4**). From the graph of clustering, the result actually surprised us a little bit that the similarity between Mark Twain's books are much lower than we thought before. We concluded that it may result in Mark Twain's different experiences during various periods of his lifetime, which not only changed his writing themes but also altered his writing style. And also, even if the authors were writing for the same theme (just like *The Adventures of Tom Sawyer* and *Alice's Adventures in Wonderland* are both about the story of children's adventure), the differences in writing style will result in the low value of similarity. Compared to Mark Twain, the author of *Alice's Adventures in Wonderland,* Lewis Carroll had a more clear individual style between her works, which can be shown from the closeness of point 2 and 6 in the graph.

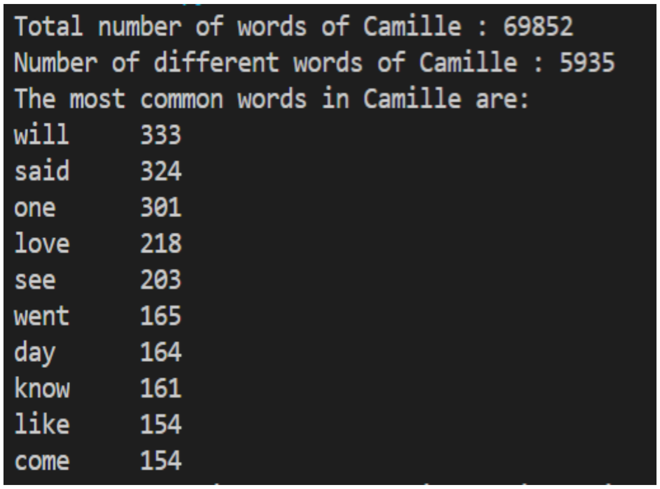
Reflection

Throughout the project, we have faced several problems when writing the code. One of the most significant issues we encountered at the beginning stage was that we had trouble printing out the words from the ebooks. We inputted the url link rather than reading the filename in our code, keeping the rest of the program unchanged. It turned out that the most common words in the books are printed out as letters instead of words. Then, we realized that the error occurred due to the data type. We struggled to convert the class type from string to io.TextIOWrapper, but we did not find a good way to achieve it. Hence, we picked another method that we decided to change the code, editing the text.split() to text.split('/n'), which means that text should be split by newlines and works successfully to show us the words instead of letters. Another issue we had was that in the clustering part, as we tried to compare different books, we noticed that the program could not read words from *A Study in Scarlet* by Arthur Conan Doyle. The program indicated the number of words in this particular book is so large that it did not have the capacity of reading it. Interestingly, there are a lot of words in Mark Twain's novels as well and the program could successfully read these texts. So here left the question that we are still confused about and could not fix the error.

As for the team process perspective, we decided to let one person from our group to work on the code. By using Zoom, we had several meetings with each other where Ziyu is responsible for writing the overall structure of the code and Diana did research online in order to tackle down the issues we mentioned early on. We decided to work in this way because if we had both of us writing code individually on our laptops, it could be messy. Once an error occurs, it would save us a lot of time if we have one person in the group to keep writing the code and the other person looks for solutions. In this way, it improved our overall working efficiency.

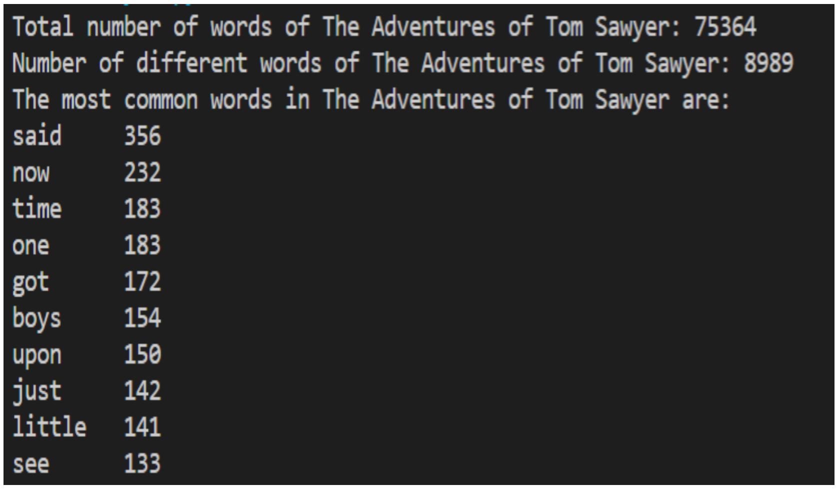
Appendix

Appendix 1. Characterizing word frequencies and the top 10 most common words in both books



```
Total number of words of Camille : 69852
Number of different words of Camille : 5935
The most common words in Camille are:
will     333
said     324
one      301
love     218
see      203
went     165
day      164
know     161
like     154
come     154
```

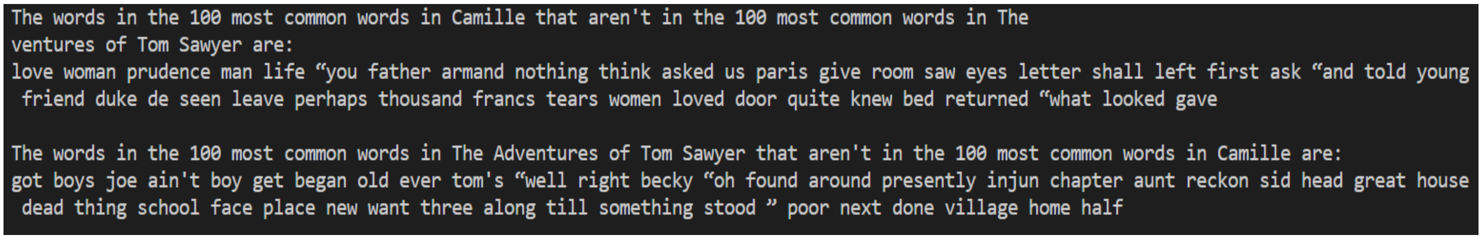
1.1 The output of Camille



```
Total number of words of The Adventures of Tom Sawyer: 75364
Number of different words of The Adventures of Tom Sawyer: 8989
The most common words in The Adventures of Tom Sawyer are:
said     356
now      232
time     183
one      183
got      172
boys     154
upon     150
just     142
little   141
see      133
```

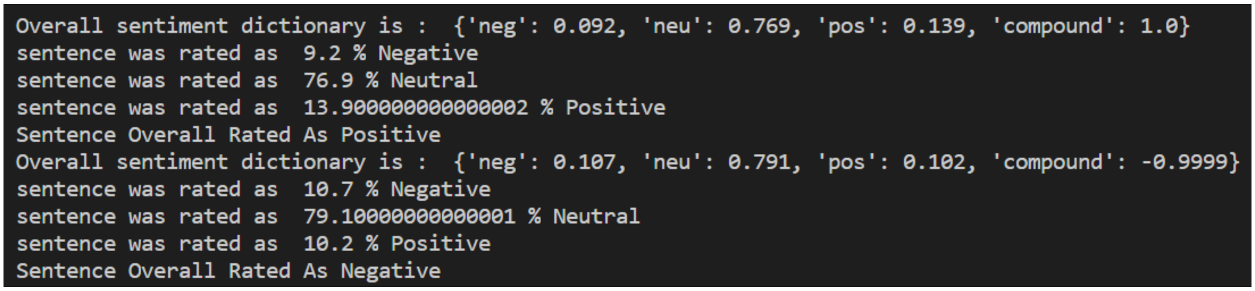
1.2 The output of The Adventures of Tom Sawyer

Appendix 2. 100 most common words in one book but are not in the other one



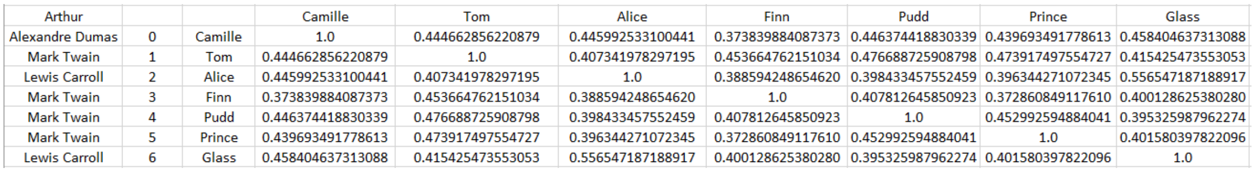
```
The words in the 100 most common words in Camille that aren't in the 100 most common words in The
ventures of Tom Sawyer are:
love woman prudence man life “you father armand nothing think asked us paris give room saw eyes letter shall left first ask “and told young
 friend duke de seen leave perhaps thousand francs tears women loved door quite knew bed returned “what looked gave

The words in the 100 most common words in The Adventures of Tom Sawyer that aren't in the 100 most common words in Camille are:
got boys joe ain't boy get began old ever tom's “well right becky “oh found around presently injun chapter aunt reckon sid head great house
 dead thing school face place new want three along till something stood ” poor next done village home half
```

Appendix 3. Sentiment Analysis result of both books



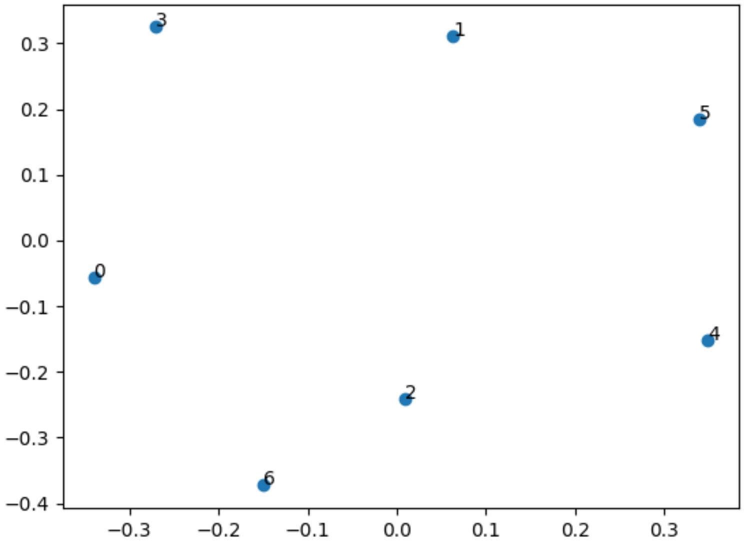
```
Overall sentiment dictionary is :  {'neg': 0.092, 'neu': 0.769, 'pos': 0.139, 'compound': 1.0}
sentence was rated as  9.2 % Negative
sentence was rated as  76.9 % Neutral
sentence was rated as  13.900000000000002 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.107, 'neu': 0.791, 'pos': 0.102, 'compound': -0.9999}
sentence was rated as  10.7 % Negative
sentence was rated as  79.10000000000001 % Neutral
sentence was rated as  10.2 % Positive
Sentence Overall Rated As Negative
```

Appendix 4. Clustering result



Arthur			Camille	Tom	Alice	Finn	Pudd	Prince	Glass
Alexandre Dumas	0	Camille	1.0	0.444662856220879	0.445992533100441	0.373839884087373	0.446374418830339	0.439693491778613	0.458404637313088
Mark Twain	1	Tom	0.444662856220879	1.0	0.407341978297195	0.453664762151034	0.476688725908798	0.473917497554727	0.415425473553053
Lewis Carroll	2	Alice	0.445992533100441	0.407341978297195	1.0	0.388594248654620	0.398433457552459	0.396344271072345	0.556547187188917
Mark Twain	3	Finn	0.373839884087373	0.453664762151034	0.388594248654620	1.0	0.407812645850923	0.372860849117610	0.400128625380280
Mark Twain	4	Pudd	0.446374418830339	0.476688725908798	0.398433457552459	0.407812645850923	1.0	0.452992594884041	0.395325987962274
Mark Twain	5	Prince	0.439693491778613	0.473917497554727	0.396344271072345	0.372860849117610	0.452992594884041	1.0	0.401580397822096
Lewis Carroll	6	Glass	0.458404637313088	0.415425473553053	0.556547187188917	0.400128625380280	0.395325987962274	0.401580397822096	1.0

4.1 Detailed data of clustering graph



4.2 Clustering graph