Text analysis with Python: Final Project
Content-based book recommendation system

György Attila Ruzicska

June 2019

1 Introduction

Recommendation systems are at the heart of many websites such as Netflix or Amazon. These systems are generally based on the users taste which is revealed by previous product searches and movie viewings, among others. Then, based on the data generated such systems can help users to be provided with more relevant information and make faster decisions.

In this project, I exploit the idea that similar recommendation systems can be built from other data sources such as books! The advantage of working with such text-heavy datasets is that we have access to a much richer resource—the whole text. My project is aimed to build the basis of a book recommendation system based on the books' content and produce statistical data analysis. The latter will be approached from two perspectives: a holistic analysis and a more specific one based on my favorite novel, the 'Les Misérables'.

2 Data description

The dataset comes from the Project Gutenberg webpage (www.gutenberg.org) which offers more than 59,000 free eBooks for download in multiple formats. Among them, I downloaded the 100 most popular books as of May - June 2019 in 'txt' format. My dataset includes well known novels such as 'The Wonderful Wizard of Oz' by L. Frank Baum, 'Pride and Prejudice' by Jane Austen and 'Adventures of Huckleberry Finn' by Mark Twain, among others. The books show great variation in the topics discussed and in the eras, they were written. In addition to the list of the top 100 books, I have also downloaded another book, 'The Odyssey' by Homer, which I will use to test the K-means and Mini Batch K-means algorithms.

I have uploaded the books to Dropbox and can be access through the following link: http://tinyurl.com/yxzepjcy

3 Text pre-processing and algorithms

As a first step, I loaded the content of each book into Python and did some basic text pre-processing. The pre-processing steps included removing non-alphabetic characters and lower casing characters. Then, I tokenized the corpus which means that I transformed each text into a list of the individual words, called tokens, it is made of. Next, I removed stop words which do not contribute considerably towards the meaning of the sentences and are generally grammatical filler words. Finally, I applied the Porter stemming algorithm in order to group together the different forms of a word so they can be analyzed as a single item: the stem.

Next, I built the bag-of-words and tf-idf models which can be used by statistical algorithms. First, I created a bag-of-words model of each texts in order to generate a universe of all words contained in my corpus. The bag-of-words model represents my books as a list of all unique tokens they contain associated with their respective number of occurrences. Second, to determine which tokens are the most specific to a book, I have used a tf-idf model (term frequency–inverse document frequency). This model defines the importance of each word depending on how frequent it is in the text and how infrequent it is in all the other documents. As a result, a high tf-idf score for a word in a text will indicate that the word is specific to that text. As a result of generating a model which associates tokens to how specific they are to each book; I could measure how related the books are to each other. To this purpose, I have used a measure of similarity called cosine similarity and visualized the results as a distance matrix, that is, a matrix showing all pairwise cosine similarities between books.

These procedures, alone, allowed me to build a book recommendation system which will be discussed further in the next section. In addition, I have also completed some general statistical analyses using dendrograms, K-means and Mini Batch K-means algorithms.

To better understand how the books in my corpus are related to each other in terms of topics discussed, I represented the whole similarity matrix as a dendrogram. For that, I used a code which computes the clusters from the similarity matrix, using the complete variance minimization algorithm. The dendrogram displays all the information about book similarities at once. For example, it shows the closest relative of each book, as well as it visualizes which groups of books have similar contents. Finally, I have applied the K-means and Mini Batch K-means algorithms to further determine how closely each book is related to one another. With clustering, I could group together books that exhibit similar properties. As a first step, I tried to determine the optimal number of clusters, however, the elbow method did not provide meaningful result. Therefore, I randomly set the number of clusters to 15. Then, I applied both the K-means and Mini Batch K-means algorithms as the latter works better with large corpuses. I have stored the results of both methods in data frames, showing the titles of the books and the cluster which they belong to.

4 General results

The general results of my project consist of two parts. First, the book recommendation program and second, the statistical data analysis.

The book recommendation program identifies the book with the highest cosine similarity to the novel that the user inputs. If the book is correctly specified, the program outputs the title of the novel and the similarity index. If the title is not correctly inputted or the book cannot be found in the corpus, the program gives an error message. Therefore, this program can be considered as a simple basis of a content-based book recommendation system. It determines the books that are closest to each other based on how similar the discussed topics are. However, it is easy to mistype the title of a book and not being aware of which books are in the corpus, therefore, an extension of allowing users to choose from a drop-down list would be indispensable.

Second, I have conducted some general analyses to reveal the overall interconnectedness of the books. The dendrogram displays the connection between all 100 books. However, because of its large size it was not possible to display the whole figure in this paper. Therefore, a smaller part of the dendrogram is displayed below. We can see that 'The Iliad' and 'Ulyssess of Ithaca' are closely related which is not surprising as they are both Greek historical novels. These books are still related but less closely to 'Leaves of Grass' and even less to 'Wit and Mirth- or Pills to Purge Melancholy'.

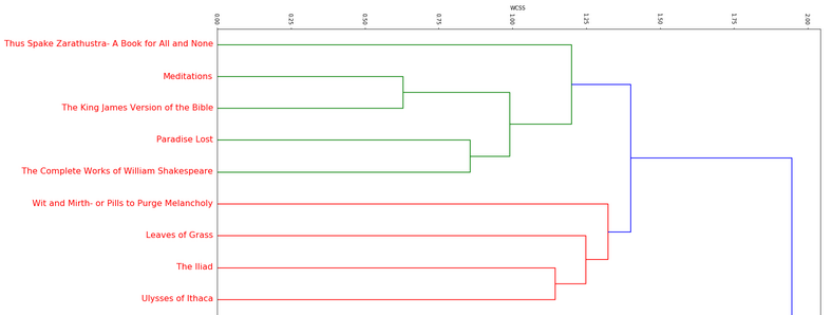

Figure 1: Dendrogram

Then, I applied clustering techniques as described above. Unfortunately, the elbow method did not provide me with the optimal number of clusters because the within group sum of squares showed a constantly decreasing trend over 40 clusters. (See Figure 2.) Therefore, I randomly chose the number of clusters to be 15.

The books and which cluster they belong to can be observed in a data frame in Python for both the K-means and Mini Batch K-means clustering techniques. To test the accuracy of these algorithms and decide which one provides better clustering, I utilized another book 'The Odyssey' by Homer.

We have seen on the dendrogram that 'The Iliad' and 'Ulyssess of Ithaca'

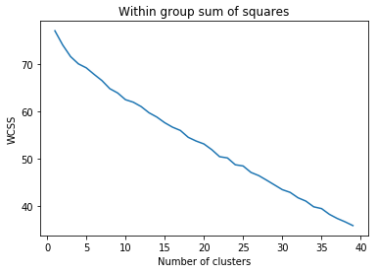

Figure 2: Within group sum of squares

are very close in content. In fact, both clustering techniques assign them to the same group. Then, it is interesting to ask whether both clustering methods assign another similar novel, 'The Odyssey', to these very same clusters. After conducting the same pre-processing steps, I could predict which cluster this novel would belong to using the previously generated K-means and Mini Batch K-means methods. From the results in Python, I have observed that only the Mini Batch K-means places the novel to the same cluster with the other Greek novels which suggests that this technique in fact works better with large corpuses.

5 Analysis of a specific novel: 'Les Misérables'

In addition to the above-mentioned results, I have conducted a short analysis focusing on a specific book. I have chosen 'Les Misérables' due to personal preferences.

First, using the tf-idf model I could determine the most frequent and unique words in the book. We can observe that the majority of these words are character names such as Valjean, Cosette and Marius.

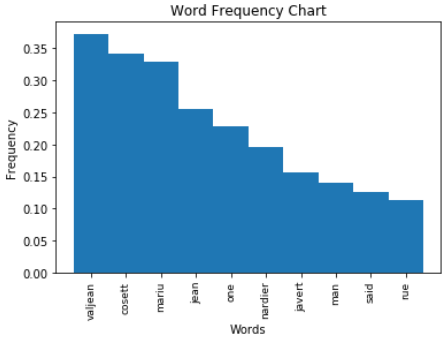

Figure 3: Most frequent, unique words in 'Les Misérables'

Second, I displayed the most similar books to 'Les Misérables'. Specifically, I produced a bar chart showing the 20 most similar books ranked by their similarity to Victor Hugo's main work. It is clear from the figure that 'Beyond Good and Evil' has the largest cosine similarity with the novel, but it is closely followed by 'The Cavaliers of Fortune' and 'Leaves of Grass'.

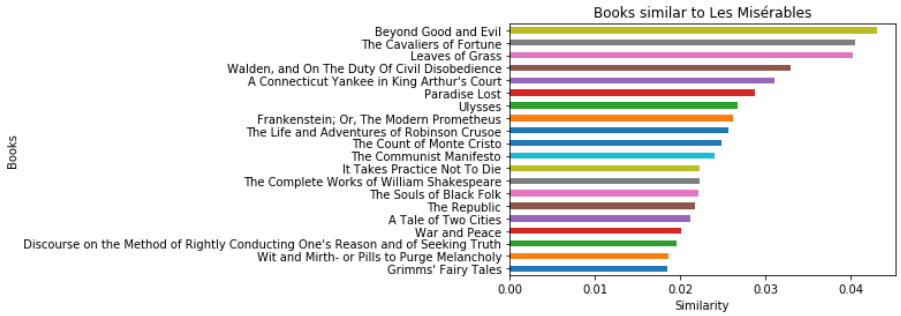

Figure 4: Books most similar to 'Les Misérables'

Finally, it is interesting to note that 'Les Misérables' has been put to the most populous clusters with both the K-means and Mini Batch K-means algorithms. However, it is difficult to determine what is the common feature of the books in this large cluster.

6 Conclusion

In this project, I aimed to build the basis of book recommendation system that may be further developed to provide a more extensive book comparison program. That is, expanding the corpus and building a better user interface could make it a well functioning and useful program. In addition, the methods that I use may have relevance in other fields and professions. For example, they can be used by text- and documents-heavy industries such as legal, tech or customer support to conduct text classification.