Text Mining of Ancient Books | Individual Coursework 2

Eyad Al-Khayat
University of Southampton
eak1u19@soton.ac.uk

ABSTRACT

The report showcases the steps taken to process and analyze twenty-four documents about the ancient past. Several data mining algorithms were used to explain how the documents relate to each other, such as K-means and hierarchical clustering. Moreover, different dimensionality reduction techniques were implemented to visualize and better understand the similarities and differences of the texts.

1 INTRODUCTION

Text mining is a valuable concept in understanding a large amount of unstructured text data, making it easier to understand and gain in-depth insight about the texts. The dataset was produced by scanning 24 history books. The work is structured in 5 sections, Section 2 goes through the steps of pre-processing and converting the data into a usable format. Section 3, presents the methods of selecting the appropriate features. Section 4, reviews and compares the results of different techniques of clustering, grouping, and visualizing the documents. Finally, Section 5, summarises the findings of the exploratory data mining techniques.

2 DATASET PRE-PROCESSING

The first step before conducting any type of analysis is to understand the data we have in hand, as certain details may give us a direction of what methods to use or avoid.

2.1 Data Characterization

2.1.1 Format. The original documents were given in the format of 24 folders containing several HTML files where each HTML file resembles a single page in the book.

2.1.2 Structure. Almost all of the HTML pages are structured in the following tree of elements:

- `Page`: A "div" element that contains all other elements.
- `Block`: A "div" element that contains paragraphs.
- `Paragraph`: A "p" element that includes the text.
- `Line`: A "span" element that contains all the words.
- `Info`: A "span" element that represent the text of each word.

This tree structure of elements will make it easy to scrap the data and turn it to a usable form.

2.1.3 Size. The total size of the HTML files is around 450 MB. This means that we will not face any issues while processing the data in memory.

2.1.4 Quality. Since the data was produced by scanning book pages using OCR (optical character recognition), there is a high possibility that the data will contain errors related to poor page scanning. Identifying and cleaning the errors manually is time-consuming, so we will use the data as it is.

2.2 Processing HTML Files

The first step was to traverse all documents one by one and go through all the HTML files in each one of them. Next, each HTML file was parsed with the help of a python package called "Beautiful Soup" that was used to extract the needed data from the HTML files with the help of the tree structure mentioned earlier. The next step was to manually extract the titles of the books from the original scans and link them back with the relevant document Id. This was useful to better understand the results of the algorithms implemented in the next sections. The output of this step was a dictionary that has the document title and its corresponding text.

3 FEATURE SELECTION

3.1 Text Cleaning

First, the text of each document was converted into lower case characters. Then, the text was tokenized into a list of words. Each word was cleaned from all types of punctuation, then, all non-ASCII words were removed. After that, the list was stripped off from all numbers, as they will not serve in having any kind of shared information between the documents. Finally, all tokens that are included in Python's NLTK stop words list were removed. Such stop words are like a, the, in, of, and,etc. Now, the text of each document is ready for the next step.

3.2 Text Vectorization

As with all data mining algorithms, features need to be described as numerical values. The process of transforming words into numbers is text vectorization. Term frequency–inverse document frequency (tf-idf) is the most popular method of giving a weight to terms in a collection of documents. The weighting of terms gives us the ability to identify terms that appear rarely in the documents.Even though, this method does not capture the true meaning of the text, however it serves as a quick and efficient way to check the similarity of documents, which is exactly what we need in our case.

4 ALGORITHMS DESIGN AND IMPLEMENTATION

4.1 Hierarchical Clustering

The reason this algorithm was implemented first is that there is no need to set the number of clusters before hand. So it will give a good indication of the documents structure. The algorithm tries to iteratively combine documents into clusters in a bottom-up approach based on some distance measure. The chosen distance measure between the documents is Cosine Similarity. Fig 1 depicts the clusters obtained by applying this algorithm on the documents.

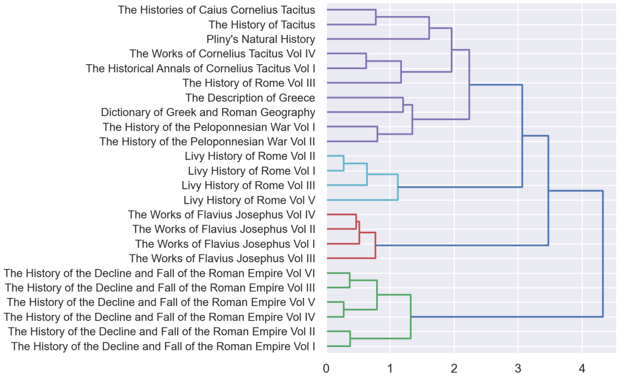

Figure 1: A Dendrogram Showing the Four Clusters Resulted from the Documents

It is clear that similar document titles were grouped together into one cluster, which means that the algorithm is clustering the documents accurately.

4.2 K-means Clustering

Unlike the previous algorithm, K-means clustering requires us to set a predefined number of clusters (k). Coming up with a suitable "k" is challenging as different "k" values will lead to different results. Fig 1 can give us an estimation of a possible good value of "k", here it is 4. However, it is clear that the fourth-top cluster can be separated into more clusters, but how many exactly? Silhouette Analysis [1] can be used as a metric to choose a "k" value that minimizes the distance between neighbouring clusters. The closer the Silhouette score is to zero, the closer the decision boundary of two neighbouring clusters are to each other. It has been found that 6 with a score of 0.247 would be a good value for "k" as after six the score is somewhat stable and is not substantially changing anymore (cluster 7 has a score of 0.249). Fig 2 shows the average score of the selected "k" clusters and the 2D visualisation of how documents relate.

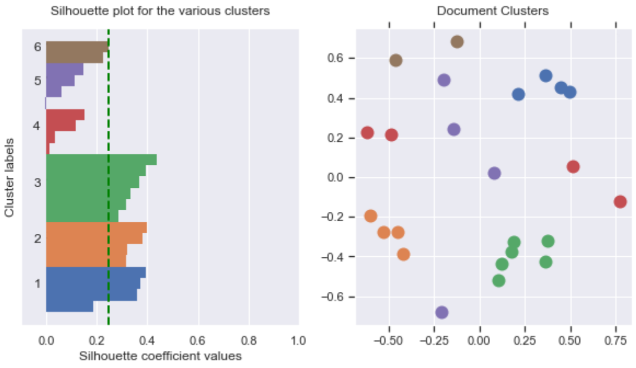

Figure 2: Silhouette Analysis score vs Document Clusters 2D Visualisation

The documents clusters visualisation was generated by reducing the dimensions of the distance matrix between clusters into 2 dimensions using Multidimensional Scaling, a technique used to visually represent distances between a collection of objects, which perfectly suites the case we have in this problem.

Table 1: Clusters Top 5 Terms

Cluster	Document Title	Top 5 Terms
1	Livy History of Rome Vol I Livy History of Rome Vol II Livy History of Rome Vol III Livy History of Rome Vol IV	appius | etruria | wherefore | licinius alba
2	The Works of Flavius Josephus Vol I The Works of Flavius Josephus Vol II The Works of Flavius Josephus Vol The Works of Flavius Josephus Vol IV	babylon | galilee | gotten | solomon | antiq
3	Decline and Fall of the Roman Empire Vol I Decline and Fall of the Roman Empire Vol II Decline and Fall of the Roman Empire Vol III Decline and Fall of the Roman Empire Vol IV Decline and Fall of the Roman Empire Vol V Decline and Fall of the Roman Empire Vol VI	danube | edit | arabian | pope | rings
4	The Historical Annals of Cornelius Tacitus Vol I The History of Rome Vol III The Works of Cornelius Tacitus Vol IV Pliny's Natural History	drusus | marius | caligula | rufus | aristocracy
5	Dictionary of Greek and Roman Geography The Description of Greece The History of the Peloponnesian War Vol I The History of the Peloponnesian War Vol II	syracuse | pausanias | attica | samos | neptune
6	The Histories of Caius Cornelius Tacitus The History of Tacitus	syracuse | pausanias | jam liber | ita |

Table 1 illustrates how the resulted clusters from K-means are quite similar to the ones shown in Fig 1. The main difference here is that we can control the number of clusters based on the problem that we have and conduct Silhouette analysis to choose the best number of clusters. It is also clear that the top terms in clusters are mostly nouns that are strictly related to the documents in that cluster, that is why it was able to categorize them quite accurately. It seems that there are two documents which are quite different than the rest of the books, the first is "The Description of Greece" and the other one is "Pliny's Natural History", these two books resulted in having them clustered in groups that may not be similar to them as shown in cluster 4 and 5.

5 CONCLUSION

In summary, although both clustering algorithms lead to similar results, k-means clustering is more robust and flexible when the number of documents increase. It is also important to keep in mind that using a different similarity measure instead of cosine similarity may result in slightly different outcomes. Moreover, advanced text vectorization could be used such as word-net, but for this problem the simple tf-idf approach did really well.

REFERENCES

[1] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* (1987). https://doi.org/10.1016/0377-0427(87)90125-7