COMP6237 Data Mining: Understanding Data

Ioan Ieremie, ii1g17@soton.ac.uk

ABSTRACT

This paper presents various data mining techniques that extract information from an unstructured dataset provided by Google. **TF-IDF** is used to vectorise the data, **K-means** is used to cluster it and **MDS** and **tSNE** are used for dimension scaling.

1 INTRODUCTION

The following report provides an insight into the flow of data with implementation details and is finished with a section of discussion of the results of the experiments. The dataset consists of 24 folders which represent 24 books that are encoded as a number of html files. These files are the result of OCR scanning.

2 IMPLEMENTATIONS & EXPERIMENTS

2.1 Data extraction

The first step is to iterate over the html files in each folder and create the content of the books. The files are analysed and the content that is the most valuable is found inside the **ocr tags** - these are extracted using the BeautifulSoup library from python. Another process is to extract the titles and authors of the books manually by looking at the entry of the book - this is necessary as there are a **large amount of misspelled words** and books with the same title.

2.2 Date pre-processing

To get only the data that is helpful we use a **regex (nltk library)** to filter all the words that do not contain any numbers of special characters. Furthermore, **stemming** the data changes all the words to their root such that we get the same meaning - this also helps with words that are misspelled at the end.

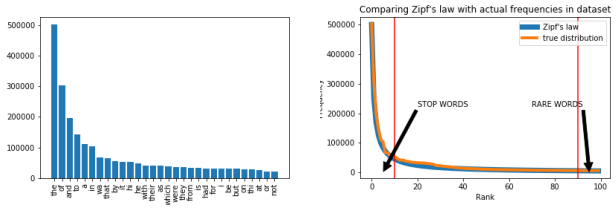

Figure 1: Words histogram

The above figures show the distribution of the first 100 most used words in the documents and it appears that it is the same as **Zipf's law**. The final dataset does not contain any **stop words** or words that appear in less that 10% of the documents - this reduces the dimensions while improving the differentiation between books.

2.3 Feature extraction

TF-IDF is used to encode the data such it can be used as feature vectors. The process of this encoding is to first create a **bag of words** containing all the prepossessed words and counting the number of times each one appears in each document - **term frequency**. Then **IDF** represents the weight of each word such that those that do not appear in most of the documents have higher weight, while those that are present in the majority of books have low weight. Here we look up **n-grams** of size 1,2 and 3 because the textbooks are about history and we want to get as much context possible. The final matrix is sparse and it has around **200000 n-grams features**.

2.4 K-means clustering

Before clustering, the **distance matrix** is computed - here we find how far two books are in the feature space by computing the cosine similarity. The distance matrix is therefore populated by the **angular similarity** which is equal to the 1 - angular distance.

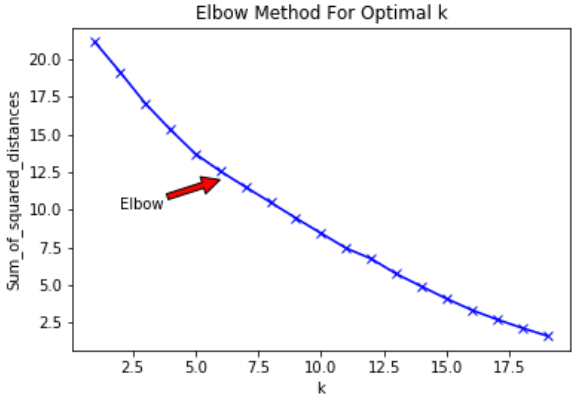

Figure 2: Elbow method

In order to find the **optimal k** for the algorithm, multiple runs are made with k in the range of [1,20]. In the experiment shown above we can get an idea about the ideal k by looking for the **elbow effect** - The plot resemblance as an arm and the optimal k is where the "elbow" can be spotted. The performance of each k is determined by the **inertia/within-cluster** sum of squares criterion.

2.5 MDS and tSNE dimension reduction

In order to see the clustering formed with the **optimal k = 6**, we need to transform the data into 2 dimensions. Because we are dealing with text, MDS gets as input the angular distance and not the euclidean distance in order to get rid of the magnitude of the features. In the scatter plot we can see that it manages to correctly position all the points in a 2 dimensional grid according to their cluster labels.

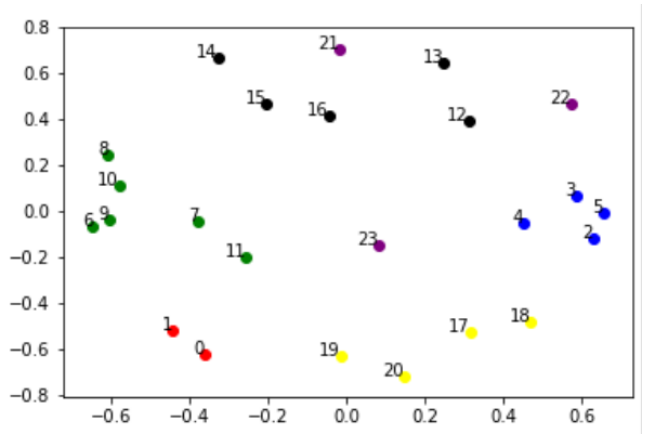

Figure 3: MDS

On the other hand, **tSNE optimises the distribution** of data such the projected data in the low dimensional space is close to the actual distribution. The experiment shows that it is not good at preserving the distance which is so valuable in our case.

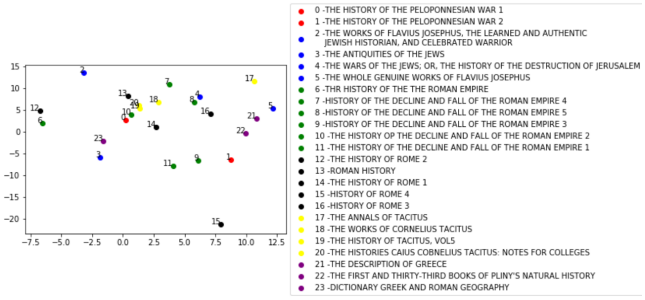

Figure 4: tSNE

2.6 Hierarchical clustering

For this part, various techniques of Hierarchical clustering have been explored. Hierarchical clustering is a way of creating a binary tree that recursively groups pairs of similar items or clusters and the results can be displayed using a **dendrogram**. Various functions for calculating dissimilarity exist but the most successful in our case are **single-linkage and complete-linkage**. The input to the algorithm is the same distance matrix used for MDS ans tSNE (angular similarity).

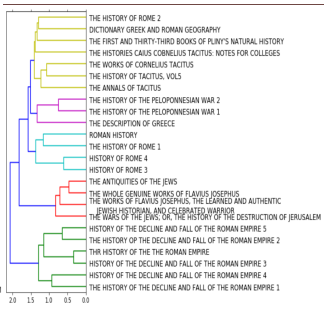

Figure 5: single-linkage **Figure 6: complete-linkage**

3 DISCUSSION AND CONCLUSION

It is interesting how these data mining techniques manage to "understand" this unstructured data, but there are a few points that must be discussed. Even if the "elbow" method is used to find the optimal k, it is hard to decide whether to choose 5 or 6 as the curve is not the evident. From the experiments, k-means converges on a good solution for the value 6 when the number of features is limited to 150000. We assume that this way, the algorithm manages to generalises better the data, which is quite sparse in our case. This is quite interesting as it finds a sixth category of books by looking at less information.

When it comes to dimension reduction, MDS performs well and it shows the clusters as expected. What can be said here is that the purple cluster formed of books 21,22 and 23 does not have a defined space. Looking at the titles, we observe that book 23 "DICTIONARY GREEK AND ROMAN GEOGRAPHY" has a high chance to present information which can be interpreted as Roman history and is found in the black cluster. Apart from all, book 22 "THE FIRST AND THIRTY-THIRD BOOKS OF PLINY'S NATURAL HISTORY" is the hardest to cluster as it does not contain any information that relates to the others. We assume that it contains information about natural history that is found around the Greece territory.

tSNE does not perform good, with no clear distinction between clusters - this is due to the fact that the features we use do not hold that much context (3 n-gram max) and it finds it harder to mimic the true distribution. We strongly believe that another form of features extraction such as word2vec can bring a lot if improvement.

The **Dendrograms** show a good clustering but there is a bit of confusion appearing. In the case of both linkage methods we see the same books that are hard to display by MDS are being randomly added to different clusters. We understand that this is hard to cluster and we are pleased to see the main categories being correctly clustered.

To conclude, it is extraordinary to see these data mining techniques can find patterns in this type of data. There are 5 clear clusters with 1 that is harder to consider as 1 cluster. MDS performs much better than tSNE for dimension reduction but we believe that further work that uses word2vec method to generate feature vectors can bring a better insight into this dataset.

3.1 REFERENCES

Houghton, J., 2020.Available at: <https://github.com/JoHoughton/Data-Mining-Demo-Code-18-19> [Accessed 14 April 2020].