COMP6237 Coursework2: Understanding Data

Thanadon Fuengworatham, Student Id: 29392802, Email: tf2n17@soton.ac.uk

***Abstract*—The aim of this coursework is to use various data mining techniques to extract insight from the provided unstructured dataset. Two main feature extracting techniques, TF-IDF and Doc2vec are experimented and compared along with clustering and dimension scaling techniques to find the optimal model to best explain this dataset.**

I. Introduction

This report is separated into two main sections. First section is to detail all the experiments and implementation of data mining techniques step-by-step following the data pipeline. After that the second section will present the result and analyse the insight from the data.

The data provided comes from OCR-scanned books consisting of 24 folders, each one contains hundreds of HTML files which represent the contents in each book. Overall, the scanned texts are mostly misspelled and contain a lot of invalid characters. Thus, all data pipeline processes are required. All the codes are implemented using Python 3.6.

II. Experiments and Implementation

A. Data Extraction

Beautiful Soup, a Python library for extracting information from HTML and XML tags, is used in this process. First of all, the inspector tool from Chome is used to find the element that store text contents. As a result from the inspection, we found that all HTML files store each word under class "ocr_cinfo", which can be acquired by HTML-parser object provided by Beautiful Soup function, $BS(open(file_path), "html.parser").select(".ocr_cinfo")$. In short, the code iterates through HTML files in 24 folder, extract text and store in the $pandas.dataframe$ object to be ready for processing. In total, 24 rows with 1 string column are created.

B. Data Pre-processing

NLTK library is utilised for cleaning the text. Firstly, RegexpTokenizer object is used with regular expression "$[a-zA-Z]+$" to extract only the words excluding all numbers and special characters. Next, words in different forms need to be unified. Stemming have advantage over lemmatization for this data as mostly the words are misspelled, where ability to maintain word meaning from lemmatization is not necessary. Additionally, some misspellings can be handled by stemming as it ignores some ending characters, which might be an incorrect part. Consequently, it can correctly unify the word and reduce feature dimension. After that, to keep only meaning words for document comparison and reduce number of text features, English stop words and words that appear less than 10% and more than 90% are removed.

C. Feature Extraction

1) TF-IDF: Sci-kit learn [1] is used for TF-IDF implementation. TF-IDF consists of two main steps, first is to create Bag of words containing vocabularies and count of their occurrences. As this dataset is from history books, n-gram is set to 3 grams to capture context in sentences with the aim to better compare similarity between books during clustering process. Second step is to normalize or weight the vocabularies by comparing from their appearance across all documents. Intuitively, high weights are given for rare words with less appearance in other documents as they hold more meaning in the document. On the other hand, low weight will be assigned to common words which exist in many documents. Due to using n-gram and large text input, TF-IDF produces a high sparse matrix containing 24 document rows and almost 200,000 word features. Moreover, although n-gram technique is opt-in to capture data context, it is not fully capable for learning all of the context from the contents. To improve this, an experiment is conducted for Doc2vec, which focuses on preserving document context.

2) Doc2vec: Gensim library [2] is used to implement this method. Doc2vec applies word2vec technique to train the neural networks not only to learn words but also the document identity, which is embedded as additional input. This technique enables an ability to compare similarity between documents rather than just words. As a result, similar documents would be vectorized and placed closely in the new vector space. As per the configurations, training epoch and output size, which collected from the weight of the neurons in hidden layer, need to be tuned. To produce the best vector output, testing has been arranged with different parameter starting from a hundred to 300 output vectors with 100 epochs as suggested in the paper [3]. Finally, the optimal configuration is 300 output vectors with 100 epochs.

D. Clustering and Visualisation

Once feature vectors are created from feature extraction step, the books data will be clustered to find the association between books. The extracted vectors from Doc2vec can be processed without problem but for features produced from TF-IDF, multi-dimension scaling technique needs to be applied to reduce dimension prior to performing clustering as it contains sparse and too large dimensions.

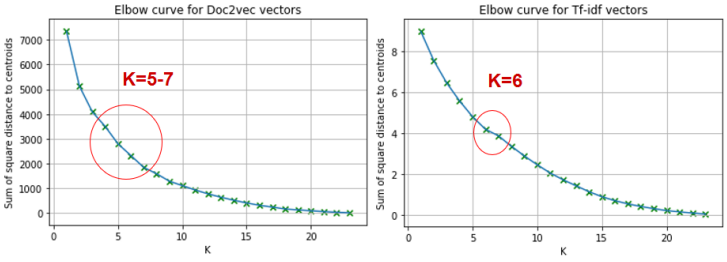

Fig. 1. Elbow method - Doc2vec (left) and TF-IDF (right)

1) Hierarchical Clustering: Scipy library is used to implement Bottom-up or Agglomerative clustering. There are several methods for dissimilarity calculation including single, complete and UPGMA linkage for distance-based and WPGMC and UPGMC

for centroid-based. To select the best method, Cophenetic correlation is used to measure the distance between the linkage data and original data, where the minimum changes of from mapping is preferred. According to the experiment, average linkage or UPGMA method produced the best score. Thus, UPGMA is used to calculate the distance for hierarchical clustering and then the result will be plotted as dendogram as shown in Figure 3.

2) K-mean Clustering: Sci-kit learn is used for this implementation. To choose the most suitable K for K-mean clustering, we use a technique called Elbow method. This method evaluates the best K which significantly reduces distance between the data points to the centroid of the cluster they belong to, which is calculated from $\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_j - \mu_i||^2)2$. The preferred K can be observed from the point on elbow shape as shown in Figure 1. The experiment has been tested in both feature from TF-IDF and Doc2vec and the results suggest that the best K should be around 6.

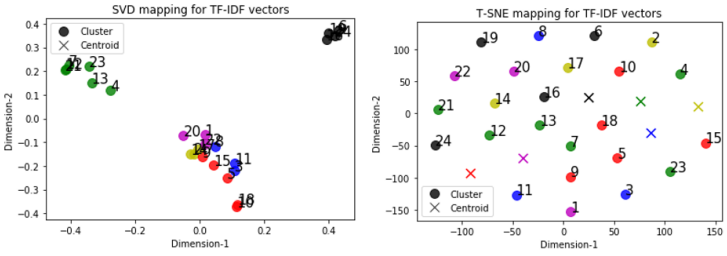

Fig. 2. Truncated SVD and T-SNE mapping results with K-mean clusters

3) Dimension Reduction: Sci-kit learn is used for this implementation. Three dimension reduction methods, MDS, T-SNE and Truncated-SVD have been experimented in this section. For MDS, as text data is likely to have different magnitude, Cosine-similarity is used rather than Euclidean to compute the distance as it is not affected by differences in magnitude. As a result in Figure 4, all groups, which are clustered in high dimension, are correctly mapped to 2 dimension space. Conversely, the mapping from T-SNE cannot preserves the distances which is sensible as its main aim is to maintain distribution of data. Truncated-SVD has been tested as its capability to well handle sparse data as presented in TF-IDF vectors. It can reduce dimension by SVD decomposition without a need of dissimilarity function. The result in the left plot on Figure 2 shows that groups with further distance from other groups are well separated; however, it cannot distinct the group of which distances are close together. In summary, the best dimension reduction method for presenting this dataset is MDS as reflected in the visualisation.

III. Analysis and Conclusion

This section is to analyse and compare the results from four selected models, Doc2vec and TF-IDF with K-mean and Hierarchical clustering, all of which use MDS as the dimension reduction method. Overall, TF-IDF models can correctly group all the books with the same series and authors in the same cluster, while some books in the highlighted rows are ambiguous for Doc2vec models. Firstly, it is interesting that Doc2vec with Hierarchical clustering groups "HISTORY OF TACITUS", which infers to an empire in Roman, with the same cluster as "HISTORY OF ROME" as shown in Figure 3. This implies that Doc2vec model might capture some similarities between context of this two books as both books are related to Roman age.

Book Id	Book Title	Doc2vec		TF-IDF	
		Hierarchical	Kmean	Hierarchical	Kmean
4,7,12,13,21,23	HISTORY OF THE DECLINE...	0	0	0	0
6,16,19,24	From Author: FLAVIUS & WHIST	2	2	2	2
3,8,11	HISTORY PELOPONNES/GREECE	5	5	5	5
2,14,17	HISTORY/ANNALS OF TACITUS	3	3	3	3
9,10,15,18	HISTORY OF ROME by geo&gor	3	4	4	4
5	HISTORY OF ROME by Theodor	1	0	4	4
20	THE FIRST AND THIRTY-THIRD	1	0	1	1
1	DICTIONARY GREEK AND ROMA	1	0	1	1
22	THE HISTORIES CAIUS COBNELI	4	1	1	1

Fig. 3. Cluster result from Doc2vec and TFIDF models

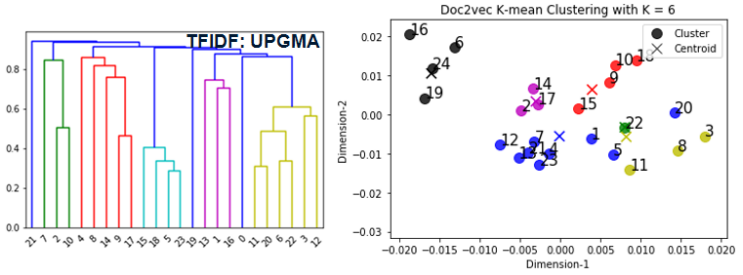

Fig. 4. Hierarchical and K-mean visualisations from TF-IDF and Doc2vec respectively

Secondly, Book 5, in the red row, which is in the "HISTORY OF ROME" series, is incorrectly grouped into other series by both Doc2vec models, while TF-IDF can correctly group it. This shows that TF-IDF can better present the dataset. Nevertheless, it can also be presumed that the context and writing styles in Book 5 are similar to Book 1 and 20 to some certain extent as Doc2vec can well capture the context of documents. Lastly, Book 1, 20 and 22, which are not series books, are distinct from other books which can be noticed from the cluster in blue line on Figure 4, where the distance, presented in Y axis, is highest compared to other clusters. Particularly, Book 22 clearly stands out from the others as presented by the K-mean plot of Doc2vec in Figure 4 where Book22 is deserted in individual cluster with out other members.

In conclusion, the books can be clustered into 5 explicit groups , where all the books are from the same series, and 1 ambiguous group consisting of Book 1, 20 and 22, where 22 is the most distinct book. For the data mining techniques, firstly, MDS can best map the data to low dimension space with good preservation of distance from original space. Secondly, both there are no significant difference from the results produced by K-mean and Hierarchical clustering. Finally, extracting feature by TF-IDF can best explain this dataset according to the sensible clusters compared from the book titles and authors. On the other hand, although Doc2vec model cannot correctly distinguish all the books, it shows some clues about connection between those books, which is interesting to do further analysis.

References

[1] "scikit-learn: machine learning in python — scikit-learn 0.19.1 documentation," http://scikit-learn.org/stable/, (Accessed on 03/16/2018).

[2] "gensim: models.doc2vec – deep learning with paragraph2vec," https://radimrehurek.com/gensim/models/doc2vec.html, (Accessed on 03/16/2018).

[3] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053