Understanding Data

ChangKai Fu, ID: 30858372, Email: ckf1n19@soton.ac.uk

***Abstract*—The purpose of this coursework is to establish a data mining process pipeline that starts from parsing the data, data pre-processing using NLP techniques, feature extraction using TF-IDF and Doc2vec, dimension reduction and clustering using Kmenas and Hierarchical clusterings. The performance of word embedding and Latent Semantics Analysis (LSA) are be compared to see which model is able to find out more correlated relationships from the textual data.**

I. Introduction

The report shows the process that how data mining technique could address the classification task of the topics of old English books scanned from OCR tools even though the source of the data is noisy as many of words are misspelled during the process.

The most interesting think in this report is to choose the right dimension reduction and verify which model is a good fit for the data by visualizing the results

II. Parsing Data

Beautiful Soup is famous for parsing HTML and XML tags. In this process, it is used to extract the content of 24 books stored under HTML tags and tranfoorm them to *panadas.DataFrame* object. During the process, all the files are found to be under class *"ocr_cinfo",* which can be selected and parsed by *html.parser* provided by Beautiful Soup. Briefly speaking, the HTML files are parsed layer by layer from external to internal document files by using Beautiful Soup and os.walk. The extracted files are all transformed to 24 rows and each row has only one column, which contains all the textual data per book.

III. Data Pre-processing

The intention in this process is to de-noise the data and tokenize the text to vectors which can be processed by computer.

NLTK RegexpTokenizer is utilized to address punctuation marks and special symbols such as " >" " =" and "■". As the data has many misspelled words, lemmatization is implemented instead of stemming since the former's percentage of words correction using *spellchecker* is much lower than the latter. Furthermore, different words might have the same stem which will lower the accuracy of the classification result. In addition, English stop words are utilised to eliminate words that appear too common.

Type of preprocess	Total words	% of correction
Stemming	3,263,370	> 20%
Lemmatization	3,147,401	7-10%

Table 1. The percentage of correction calculated by spellchecker

`

IV. Feature Extration

TF-IDF and doc2vec are chosen as feature extractors in this process. The combination of TF-IDF and cosine similarity is a standard selection method for latent Semantic Analysis (LSA). By comparison, doc2vec is a documentation filtering version of word2vec proposed in 2014[1], which is a popular and critical feature extractor using word embedding technique for document contains many sentences.

1) TF-IDF: The model is using Bag-of-words to calculate the term frequency appearing in the document and inverse term frequency (IDF) to keep the keywords in the corpus. The output of TF-IDF is a high dimensional sparse matrix containing 24 rows and 219,525 features. The matrix is then processed by 1-cosine_similarity(matrix) so that the data can be represented by a 24 x 24 matrix with the similarity value between zero and one. The advantage is that it takes a lower computing cost but the disadvantage is also obvious that the two words with similar meaning are unable to be represented by the model

2) Doc2vec: Word embedding technique is created to solve the above problem, which is also the main reason that it is chosen to implement the task. As the document filtering version of word2vec, doc2vec model generates the weighted matrix in the hidden layer after the model is trained by the entire corpus, which means the textual data is transformed to a new vector space that the documents with similar topic will have closer vectors. To have a more accuracy clustering result, the size the processed vectors are 300 with epoch equals to 100 as this is the standard configuration for doc2vec according to the essay[2]

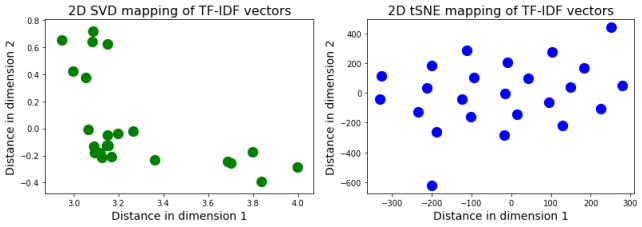

Figure 1. 2D SVD and tSNE mapping of TF-IDF vectors

V. Dimension Reduction

t-distributed stochastic neighbor embedding (tSNE), Principal Component Analysis(PCA) and Multi-dimensional Scaling(MDS) are tested to have a clear insight which tool gives a better performance for clustering using two-dimensional visualization.

For the vectors processed by TF-IDF and cosine similarity, the ideal tool should be able to generate a non-Euclidean

dimension reduction as the result of cosine_similarity does not includes the magnitude of textual vectors. MDS becomes the best candidate in the part as the nonmetric configuration of MDS satisfys the need. On the other hand, the distribution of SVD mapping is very crowded, which discloses that the eigenvectors might not represent the difference between books. In comparison, tSNE distributes evenly in the space, which makes clustering difficult.

The size of doc2vec processed vectors is 24x400, which makes tSNE and MDS unable to be utilized since they require a symmetric input matrix. Therefore, SVD is opted to reduce the features for doc2vec model.

VI. Clustering and Visualisation

1) K-means Clustering: The model ends when the least summation of distance between each data and the centroids are fixed after numbers of iteration. Since this is an unsupervised learning, elbow curve and Silhouette score are jointly considered to give K equals to six for TF-IDF and the value for doc2vec is five or six as demonstrated in figure 2 and 3.

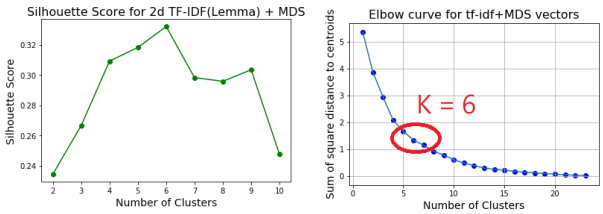

Figure 2. Elbow curve and Silhouette score of TF-IDF vector

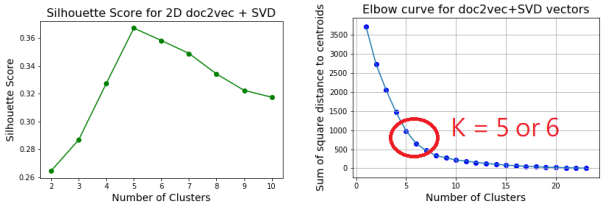

Figure 3. Elbow curve and Silhouette score of doc2vec vector

2) Hierarchical clustering: In this process, the average configuration of agglomerative clustering is opted after comparing to the other three, i.e. single, complete and ward since it's Cophenetic correlations closer to one no matter the model is TF-IDF or Doc2vec. Thus, the average method is used for the calculation of distance for hierarchical clustering and the result is plotted as dendogram as shown in Figure 4.

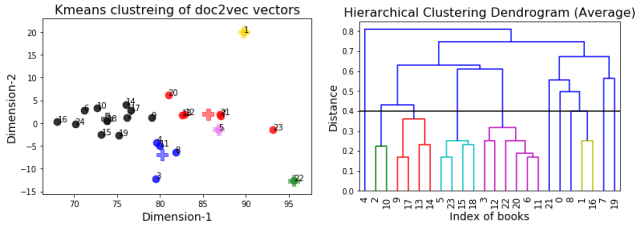

Figure 4. Kmeans clustering for TF-IDF (Left) and Hierarchical clustering for Doc2vec (Right)

VII. Conclusion

As we know this is an unsupervised learning problem, we do not 100% know if the answer is correct, but we could infer from the title of books and name of authors to know part of the possible answers since they are series books are written by the same writers as shown by Table 2.

In conclusion, doc2vec performs better than TF-IDF as more books are classified to the more likely correct groups and distinct book such as No. 5 and 22 are also well-classified. In addition, Kmeans model outputs a more accuracy clustering result for doc2vec than hierarchical clustering. However, the model does not well-split books "The history of the decline and fall of the Rome Empire" (7,12,13,21,23) "The Annels of Tacitus"(1, 15, 17) and "History of Rome"(4, 10, 14, 18)(Orange coloring in Table 2). This possible reason might be books are originated form the same age or the writing style of the writers are a bit similar. For TF-IDF, Kmeans performs better than Hierarchical clustering especially in the classification of the distinct book No. 5 "DICTIONARY GREEK AND ROMAN GEOGRAPHY"

The above result conforms to my hypothesis as doc2vec correlates the concept of documents so that similar documents can be vectorized and placed closely in a new vector space. On the other hand, I would say the clustering task for these 24 books demonstrates the shortcoming of TF-IDF as it focuses only on the frequency and inverse-frequency of words. However, the result is not ideal enough to say the model is well-trained to have an accuracy classification ability, more research papers should be read to find out proper configuration for the above mentioned models.

Book No.	Book Title	TF-IDF		Doc2vec	
		Kmeans	Hierarchical	Kmeans	Hierarchical
7, 12, 13, 21, 23	HISTORY OF THE DECLINE	V	V	V	V
6, 16, 19, 24	Related to FLAVIUS JOSEPHU	V	V	V	V
15, 17	THE ANNELS OF TACITUS.	X	X	V	V
4, 10, 14, 18	HISTORY OF ROME	V	V	V	X
5	DICTIONARY GREEK AND	V	X	V	V
22	THE HISTORIES CAIUS	X	X	V	V
20	THE FIRST AND THIRTY-	X	X	X	X
1	THE HISTORY OF TACITUS	X	X	X	X

Table 2. Comparison of the clustering results by TF-IDF and Doc2vec models

REFERENCES

[1] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).

[2] "gensim: models.doc2vec – deep learning with paragraph2vec,"