COMP6237 Data Mining CW2: Understanding Data

Hang Su, Student Id:30005019, Email:hs1a18@soton.ac.uk

ABSTRACT

In this paper, we present some unsupervised learning approaches performing on a text clustering task. Three main technologies are applied in the project, they are web crawler, feature extraction technologies and clustering algorithms respectively. Moreover, we need to compare performances of different models to find the optimal one which can cluster the books best.

1 INTRODUCTION

The overall task of this coursework is to perform data mining technologies on a data set that contains 24 texts about Antiquity. And the texts are provided as OCR-scanned books consisting of 24 folders, each folder contains hundreds of HTML files, which leads us to build a crawler to get the textual data first.

The note is organised as followings: Section 1 gives an introduction. Section 2 describes the whole procession of the experiment step by step and provides some information on the approaches used in this data mining task. Section 3 displayed some visual result of the experiment. Section 4 provides a brief analysis of the project.

2 EXPERIMENT AND IMPLEMENTATION

2.1 Crawling the Data

To grasp data from HTML tags, the Python library Beautiful Soup is generally thought as a convenient tool. First, open some HTML files by explorer and use the inspector tool of the explorer and find that each line is wrapped by a pair of span tags whose class properties are "ocr_line", and each single words are included in "span" tags. Thus it is convenient to acquire the texts through "soup.find_all('span', 'class': 'ocr_line')". Next, the biggest problem is that there are massive incorrect words in the texts. Some spell checking libraries like "autocorrect" can be used to solve a part of these problems.

2.2 Data Pre-processing

Entering this part, the main purpose was to remove the stop-words and tokenize the texts. The NLTK library was utilized to achieve these. "wordpunct_tokenize()" method can separate the texts and punctuation easily, after that, list generation can help to remain all the texts. Besides, it is obvious that there are lots of redundant single characters which may involve the final results, thus we removed all the words that are shorter than 3.

2.3 Feature Extraction

In this part, it is impossible to decide which feature extraction approach will perform better on this task, thus applying multiple methods and compare the results is a better choice. We chose TF-IDF and Doc2vec to build the feature matrix.

2.3.1 TF-IDF. TF-IDF is a very common and popular information retrieval scheme, it intends to reflect the importance of a word to a document. Generally, it contains two main steps: The first step is calculating the term frequency, which means creating a bag of words containing vocabularies and count of their occurrences. The second step is calculating the inverse document frequency, this means weighing the vocabularies by comparing from their appearance across all documents. "sci-kit learn" provides us useful tools, making it easy to implement TF-IDF. Due to setting the parameters, we filtered the words that are lower than 10 percent and higher than 90 percent as well as got a feature matrix which shape is 24 by 10000 to be used to do the clustering in the next stage. However, TF-IDF merely thinks the importance of words, ignoring the contextual information, to improve this, reconstructing the feature matrix using Doc2vec is useful.

2.3.2 Doc2vec. Doc2vec is designed based on word2vec, it is a breakthrough on embeddings. Through applying doc2vec, We can use a vector to represent a sentence or document, Mikolov et al.[2] call it as "Paragraph Vector". Nowadays, it is widely used as an unsupervised learning approach to learn document representation. The library "gensim" provide tools to implement this. First of all, we need to tag the document. Then it is a vital process to train the model and build vocabulary. Next, we can encode it and build the feature matrix to be used in classifiers[1].

3 MODEL TRAINING RESULTS AND ANALYSIS

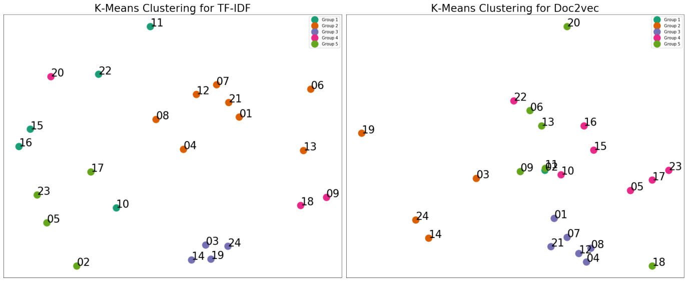
Because generated two different feature matrices, passing them to different classifiers and comparing the results is necessary. For the clustering algorithms, we choose K-means, hierarchical and mean-shift. It is worth to mention that some dimension reduction technologies like multi-dimension scaling need to be applied before mapping the data onto a two-dimensional plot. Before showing the result, this paper provides a table 1 with book_id and titles of the books.

Book_id	Title
01	THE HISTORY OF THE DECLINE AND FALL OF THE ROMAN EMPIRE. VOL. VI
02	THE HISTORIES CAIUS COBNELIUS TACITUS
03	THE WORK OF JOSEPH US, THE JEWISH WAR. VOL. IV
04	THE HISTORY OF THE DECLINE AND FALL OF THE ROMAN EMPIRE. VOL, I
05	THE HISTORY OF TACITUS. BOOK I. VOL. V
06	THE FIRST AND THIRTY-THIRD BOOKS OF PLINY'S NATURAL HISTORY
07	THE HISTORY OF THE ROMAN EMPIRE. VOL. V
08	THE HISTORY OF THE DECLINE AND FALL OF THE ROMAN EMPIRE. VOL. II
09	THE HISTORY OF THE PELOPONNESIAN WAR. VOL. II
10	TITUS LIVIUS' ROMAN
11	THE HISTORY OF ROME, BY TITUS LIVIUS. VOL. I
12	THE HISTORY OF THE DECLINE AND FALL OF THE ROMAN EMPIRE. VOL. IV
13	DICTIONARY GREEK AND ROMAN GEOGRAPHY. VOL. II
14	THE LEARNED AND AUTHENTIC JEWISH HISTORIAN AND CELEBRATED WARRIOR. VOL. III
15	LIVY. VOL. III
16	LIVY. VOL. V
17	THE HISTORICAL ANNALS OF CORNELIUS TACITUS. VOL. I
18	THE HISTORY OF THE PELOPONNESIAN WAR. VOL. I
19	THE LEARNED AND AUTHENTIC JEWISH HISTORIAN, AND CELEBRATED WARRIOR. VOL. IV
20	THE DESCRIPTION OF GREECE
21	THE HISTORY OF THE DECLINE AND FALL OF THE ROMAN EMPIRE. VOL. III
22	THE HISTORY OF ROME. VOL. III
23	THE HISTORY OF TACITUS. BOOK I. VOL. IV
24	THE FLAVIUS JOSEPHU

Table 1: Book List

3.1 K-Means Clustering

The clustering results are shown in Fig 1 and Table 2. It is obvious that there are only some tiny differences between the clustering result of applying these two approaches. Through searching the synopsis of these books, we find lots of books all contain contents about Roman and Greece besides books 03,14,19,24 which describe the culture of Jewish, so it is reasonable to see that there are some changes when applying different feature matrix.



(a) TF-IDF (b) Doc2vec

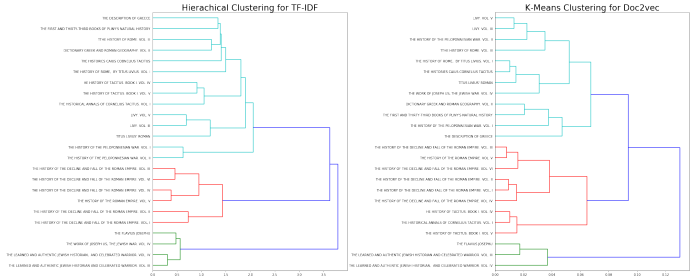
Figure 1: MDS mapping for K-Means

Approach	Label	Book_ID
TF-IDF	Group 1	10,11,15,16
	Group 2	01,04,06,07,08,12,13,21
	Group 3	03,14,19,24
	Group 4	09,18,20
	Group 5	02,05,17,23
Doc2vec	Group 1	02
	Group 2	03,14,19,24
	Group 3	01,04,07,08,12,21
	Group 4	05,10,15,16,17,22,23
	Group 5	06,09,11,18,20

Table 2: K-means Result

3.2 Hierarchical Clustering

When apply hierarchical to do the clustering, the texts are only separated into 3 groups, but we thought it is more reasonable, the reason is the same as claimed in subsection 3.1.



(a) TF-IDF (b) Doc2vec

Figure 2: Hierarchical Clustering Result

Approach	Label	Book_ID
TF-IDF	Group 1	02,05,06,09,10,11,13,15,16,17,18,20,22,23
	Group 2	01,04,07,08,12,21
	Group 3	03,14,19,24
Doc2vec	Group 1	02,03,06,09,10,11,13,15,16,18,20,22
	Group 2	01,04,05,07,08,12,17,21,23
	Group 5	14,19,24

Table 3: Hierarchical Result

3.3 Mean-Shift Clustering

When implementing the experiment, this method only can be applied on doc2vec matrix because the tf-idf matrix is sparse. But it is disappointing that the result is worse than the previous. Group 3 and Group 5 even can be regarded as outlier values.

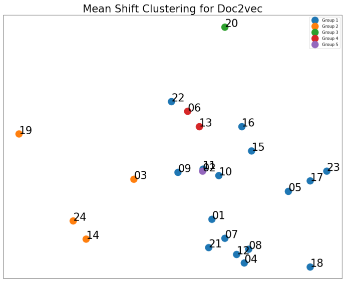


Figure 3: Mean-Shift for Doc2vec

Approach	Label	Book_ID
Doc2vec	Group 1	01,04,05,07,08,09,10,11,12,15,16,17,18,21,22,23
	Group 2	03,14,19,24
	Group 3	20
	Group 4	06,13
	Group 5	02

Table 4: Mean-Shift Result

3.4 Latent Dirichlet allocation

We also try to use LDA to output the topics of the documents, we set the model to give 5 topics and each topic contains 30 words, the result is shown below. Unfortunately, we can only draw a little information. In the future, working more on data cleaning may help us acquire better result.

0	0.005*""lib" + 0.004*""decline" + 0.003*""justinian" + 0.002*""constantinople" + 0.002*""church" + 0.002*""belisarius" +
1	.008*""athenians" + 0.003*""lucius" + 0.003*""nero" + 0.003*""peloponnesian" + 0.003*""dictator" + 0.002*""tacitus"
2	0.005*""cf" + 0.005*""strab" + 0.003*""pp" + 0.003*""pliny" + 0.002*""ptolemy" + 0.002*""liv"
3	0.005*""homer" + 0.004*""fay" + 0.004*""olympic" + 0.003*""statuary"
4	0.020*""jews" + 0.009*""herod" + 0.008*""josephus" + 0.006*""jerusalem" + 0.006*""antiquities" + 0.003*""david"

Table 5: LDA topics

4 DISCUSSION

In conclusion, it is hard to say how many categories to divide these books into is better because there are several books contains similar contents. However, we also can draw a conclusion that book 03,14,19,24 always belong to the same series cause they all describe information about Jewish. For other books, they depict similar information, thus they can be classified into the same group. However, according to LDA analysis result, they sometimes can be divided into different groups may base on the authors of the books. Furthermore, though the Doc2vec method does not show much better performance in this task, it still can show some relationship between those books, which is valuable to do further analysis on it.

REFERENCES

[1] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).