Data Mining COMP6237 – Individual coursework

Nicola Vitale
MSc Data Science
nv5g15@soton.ac.uk

ABSTRACT

In this paper the experience of understanding unstructured data is reported. The coursework, as part of the Data Mining module, involved the understanding a series of texts data using various techniques.

1. INTRODUCTION

The data provided consisted of a corpus of 24 books. For each book we have the output of OCR processing: the content of each page has been provided as an html file.
The approach used involved two main phases:

- Data preprocessing
- Understanding the data

In the first phase the effort was aimed at obtaining the raw data by extracting the raw text and finding an appropriate data structure to work with Python. In the second phase various data mining techniques have been applied to the data corpus in order to understand its hidden structure.

2. DATA PREPROCESSING

Initially the text of the books has been extracted from the html file. For this purpose, it has been convenient to use the os library to iterate through the files and the re library to remove html tags keeping the text of each page. A regular expression to identify each tag has been found and once the page was cleaned the text of the book was recreated and saved as a txt file. The output of this operation consisted in 24 txt files corresponding to the books texts.
Then it has been decided to work with two main lists in Python: a list of 24 texts and a parallel list with the corresponding titles. These data structures have been saved using pickle library. In general over the project this library has proved to be very useful since it allowed to save the output of each analytical phase.

3. UNDERSTANDING DATA

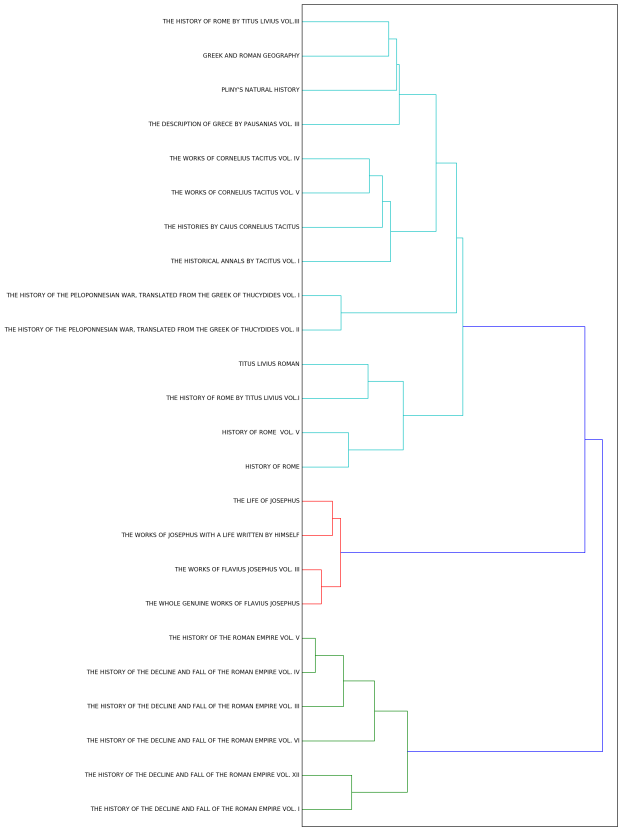
3.1 Vocabularies

In understanding data the pandas library has been massively used to create useful data frames to access the data.
A data frame contained the whole corpus of stemmed words as rows and the tokenized words as columns has been created. This in order to use the stemmed books in the algorithms but being able to access the corresponding token when needed. The data frames contain 5,618,845 items. Words have been repeated and a stem could clearly correspond to many tokens.

3.2 Tf-idf matrix and cosine similarity

Iterating through the temmed books list a tf-idf matrix has been created using the TfidfVectorizer function in the sklearn module. This function allows us to input the tokenizer function previously written and to set the parameters that regulate the dimension of the matrix.
From tf-idf the cosine similarity between documents can be computed using the cosine_similarity function in the same module. In the following algorithms a distance matrix has been used as more practical. This can be obtained just doing: $Distance\_matrix = 1 - Cosine\_similarity\_matrix$.

3.3 Hierarchical clustering



3.4 Kmeans clustering and multidimensional scaling