CS-E4650 Methods of Data Mining
Project work

Bernard Spiegl (1023112)

1 Introduction

This project covers the process of text clustering from data preprocessing to performing dimensionality reduction and finally clustering the data and evaluating the obtained clusters. The methods used for preprocessing, clustering and dimensionality reduction are covered in section 2. Afterwards, we go over results in section 3. Finally, some instructions about the project, required libraries and code are given in section 4.

2 Methods

This section covers all the methods used for performing the analysis for the project.

Preprocessing

In order to perform necessary text preprocessing I make use of `nltk`[1] Python library. First the title and abstract are combined. Thereafter, stopwords are eliminated from the text and stemming or lemmatization is performed, the stopword list used is the english list contained in the `nltk`. In order to perform stemming I use `SnowballStemmer` and for lemmatization (when performed) I use `WordNeLematizer` (although I tested both lemmatization and stemming, there was not a significant performance difference between the two so I stuck to stemming as whole context of the word is not as important for this task) Afterwards, data is transformed using `TfidfVectorizer` and normalized (`tf-idf` stands for term frequency-inverse document frequency). Tf-idf computations are performed using the following two equations:

$$\text{tf-idf(t, d)} = \text{tf (t, d)} \cdot \text{idf(t)}$$

[1] natural language toolkit

$$\text{idf(t)} = log(n/\text{df(t)}) + 1$$

where df(t) is the document frequency of t and tf is term frequency of t in d.

Clustering

For clustering I used several different approaches. Namely, K-Means with Euclidean distance, Agglomerative Clustering as well as Spectral Clustering.

Dimensionality Reduction

I used various dimensionality reduction methods such as PCA[2], t-SNE[3] and TruncatedSVD[4]. They were mainly used to perform dimensionality reduction in order to be able to visualize the clusters in 2D and 3D spaces. After running some experiments I realized that the methods didn't have a significant impact on the clustering outcome. However, they reduced the computing time needed in order to cluster the data as a result of the reduced dimensionality.

3 Results

NMI Comparison

Unfortunately, I wasn't able to achieve NMI scores close to the original 0.81 proposed in the assignment instructions. Results of all clustering methods are available in Table 1. Furthermore, visualizations of `K-Means` clustering results are available in Appendix A for `t-sne`, `PCA` and `TruncatedSVD` both in 2D and 3D spaces.

Method	NMI
K-Means	0.461
Agglomerative	0.447
Spectral	0.432

Table 1: Comparison of NMI scores using different clustering methods.

For Agglomerative Clustering linkage used was `complete` and for Spectral Clustering the affinity used was `nearest neighbors`.

[2]Principal Component Analysis
[3]t-distributed Stochastic Neighbor Embedding
[4]Truncated Singular Value Decomposition

However, regardless of a potentially suboptimal NMI scores I was clearly able to distinguish the topics covered by each of the clusters as shown in the following subsection.

Content Analysis

By analyzing plotted word scores and WordClouds available in Appendix B, we can clearly infer topics of different clusters and conclude that our data encompasses the following topics:

- cluster 0: compilers, programming, computing, etc.
- cluster 1: image detection, computer vision, object detection, etc.
- cluster 2: databases, relations, queries, data, etc.
- cluster 3: security, encryption, protocols, etc.
- cluster 4: robot, control, systems, etc.

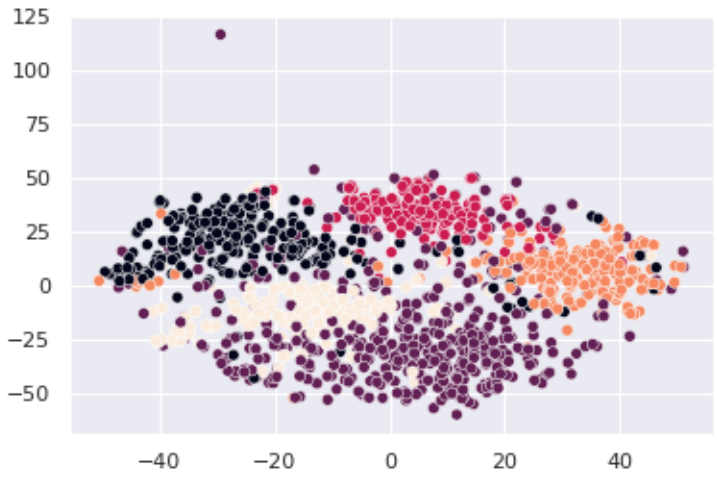
4 Instructions

The solutions are provided in a form of an interactive Python Jupyter Notebook. The required libraries are:

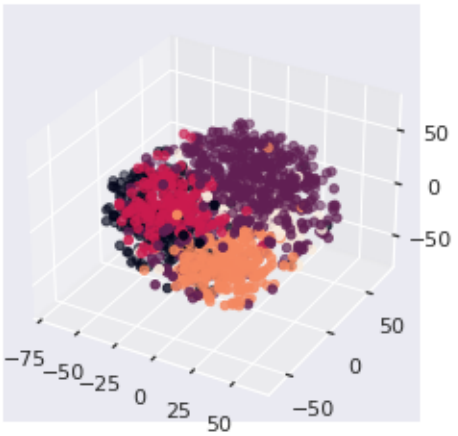
- `numpy` - used for numerical operations
- `pandas` - used to load and manipulate data
- `nltk` - used for stemming/lemmatization and stopword removal
- `sklearn` - used for performing tfidf vectorization, clustering as well as data dimensionality reduction
- `matplotlib` and `seaborn` - used for visualization
- `wordcloud` - used for making wordclouds to visualize frequency of terms

Appendices

A Cluster Plots

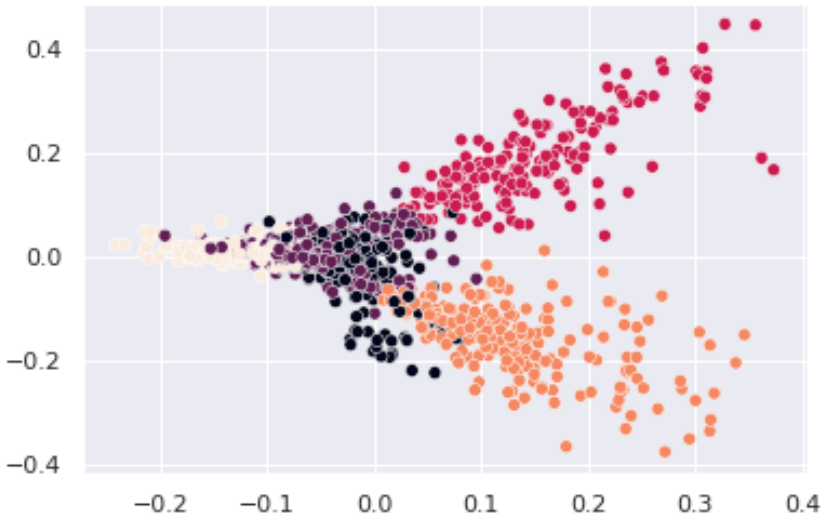


(a) 2D

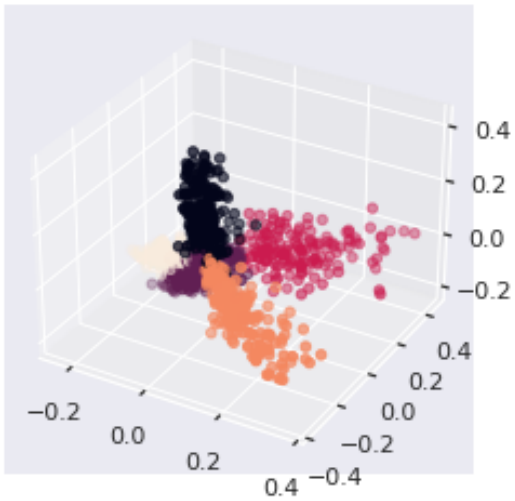


(b) 3D

Figure 1: Dimensionality reduction using t-sne.

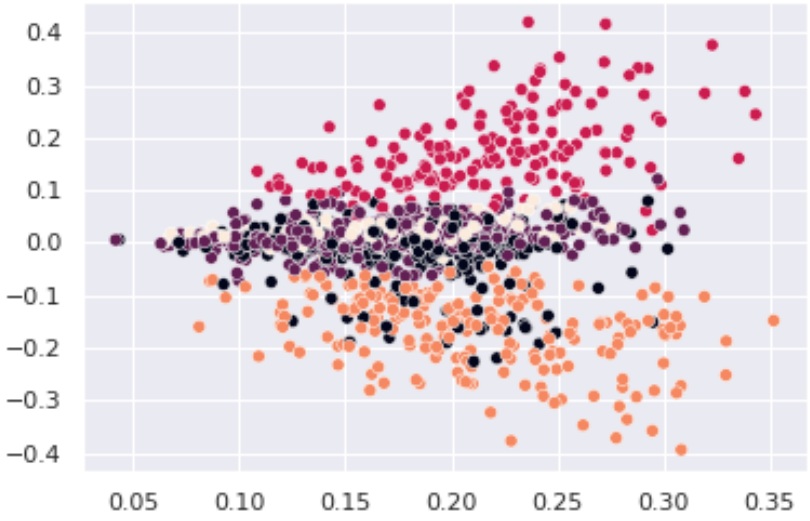


(a) 2D

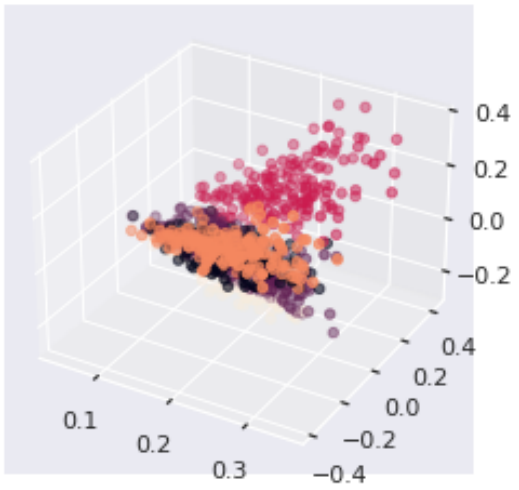


(b) 3D

Figure 2: Dimensionality reduction using PCA.



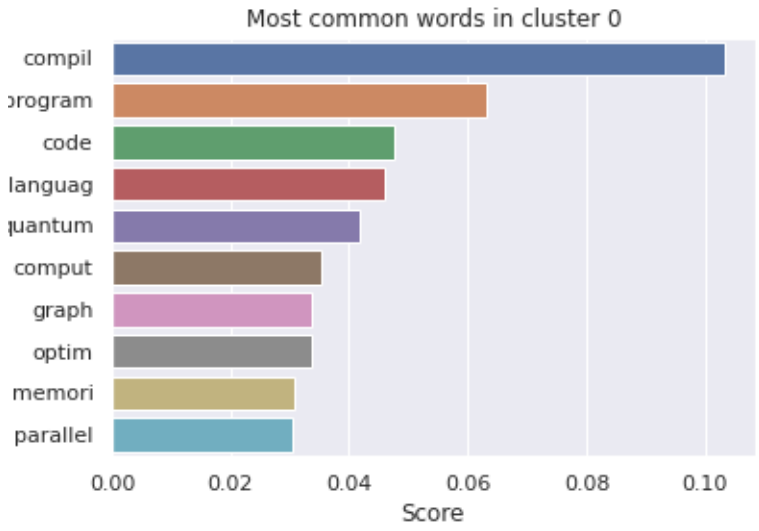
(a) 2D



(b) 3D

Figure 3: Dimensionality reduction using TruncatedSVD.

B Word Frequencies and WordClouds



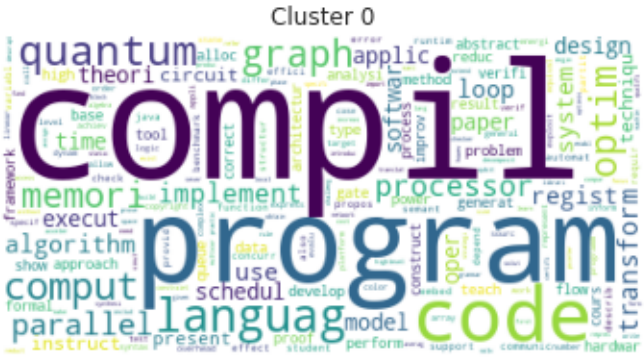


Figure 4: Cluster 0 most common words and the corresponding WordCloud.

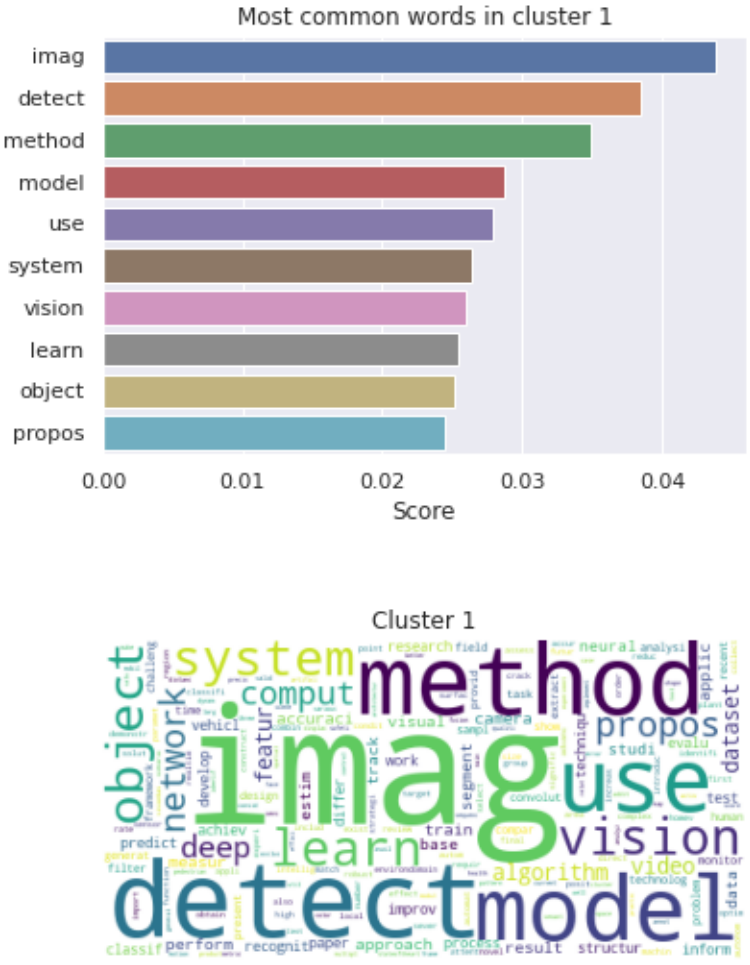
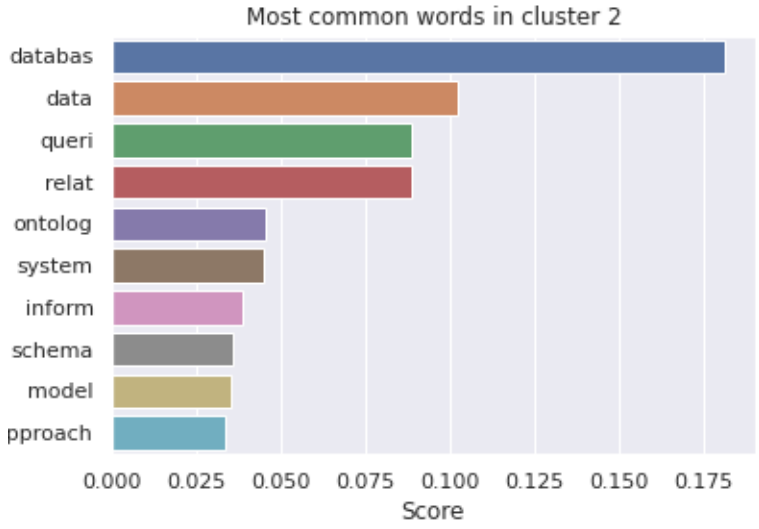


Figure 5: Cluster 1 most common words and the corresponding WordCloud.



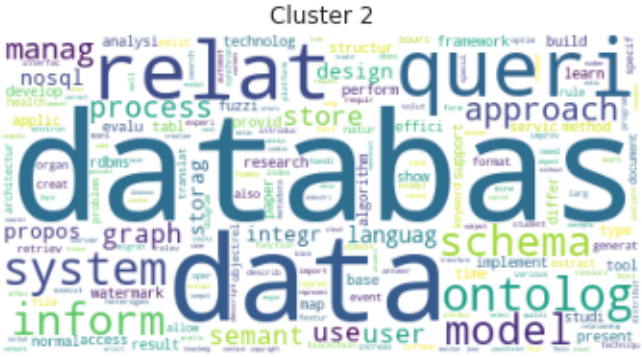


Figure 6: Cluster 2 most common words and the corresponding WordCloud.

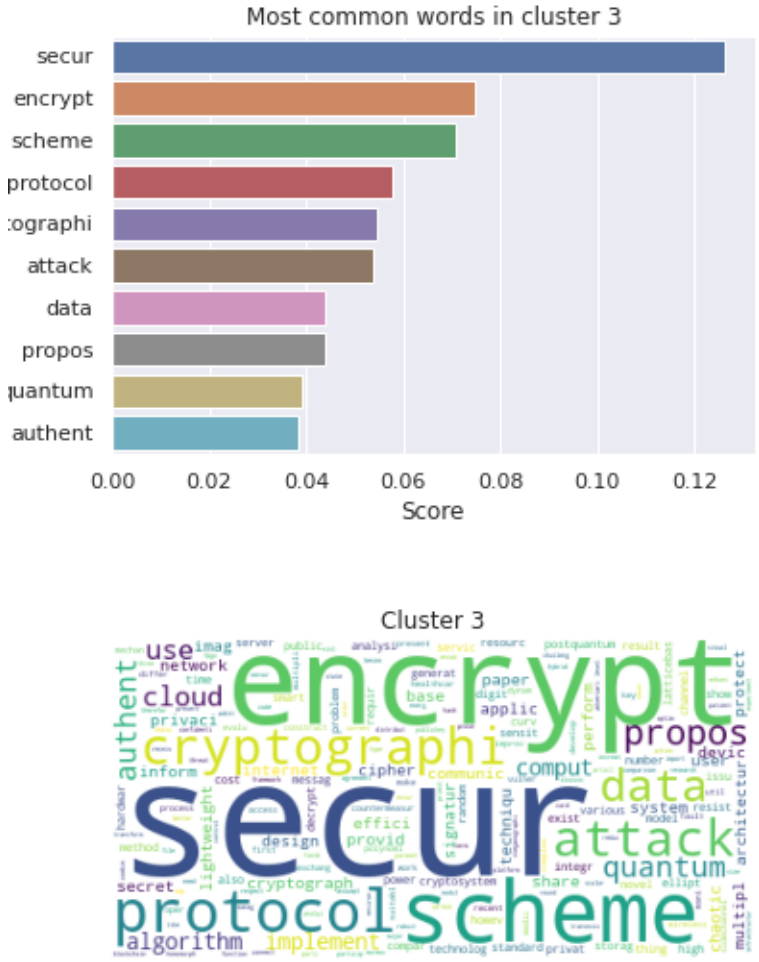
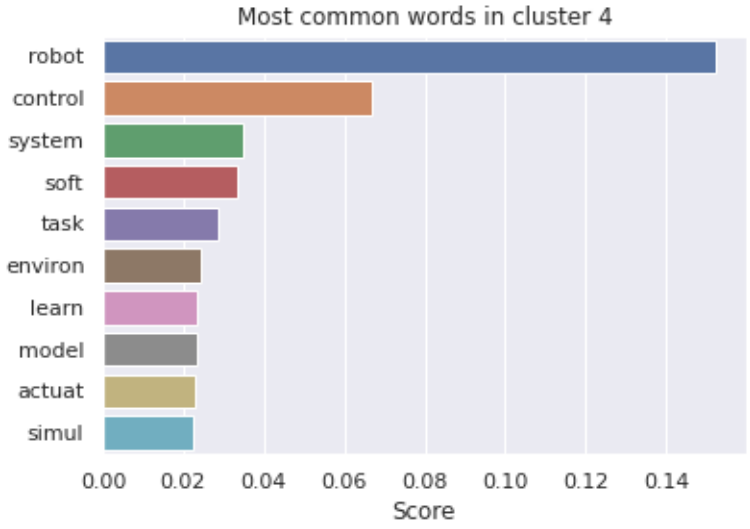


Figure 7: Cluster 3 most common words and the corresponding WordCloud.



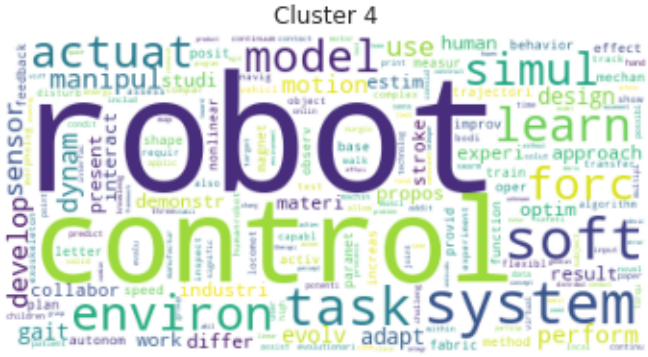


Figure 8: Cluster 4 most common words and the corresponding WordCloud.