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

Preprocessing

In order to perform necessary text preprocessing I make use of `nltk`[1] Python library. First the title and abstract are combined. Thereafter, punctuation, words containing digits and stopwords are eliminated from the text and stemming 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) `WordNeLematizer` is used. Moreover, urls are also transformed, double whitespaces removed and in order to increase performance common top words between clusters are removed as well - in my case this was a single word `"use"`. 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:

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

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

[1]natural language toolkit

where df(t) is the document frequency of t and tf is term frequency of t in d. Through experimenting and tweaking various parameters I've come to realization that in `l1` norm and applying sublinear tf scaling (tf is replaced with $1 + log(\text{tf})$) in combination with K-Means yields the best results.

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

Results of all clustering methods are available in Table 1 with K-Means achieving the highest result with NMI of 0.829 (in order to ensure reproducibility the `random_state` parameter of K-Means is fixed to 666998, this number was obtained by searching through several seed numbers and picking the best performing one) followed by Spectral Clustering with NMI of 0.805 and Agglomerative Clustering in the last place with NMI of 0.23. It is also worth noting that Agglomerative Clustering performed significantly worse than other two tested methods. For Agglomerative Clustering `linkage` used was `complete` and for Spectral Clustering the `affinity` used was `cosine` with `assign_labels` set to `discretize`.

Hereafter, I was clearly able to distinguish the topics covered by each of the clusters as shown in the following subsection.

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

Method	NMI
K-Means	0.83
Spectral	0.81
Agglomerative	0.23

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

Content Analysis

After clustering I was able to extract most important words in each of the clusters. 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: topics related to robots, systems, control, etc.
- cluster 1: topics related to object detection, computer vision, images, machine learning, etc.
- cluster 2: topics related to security, encryption, cryptography, etc.
- cluster 3: topics related to code, programs, compilers, programming languages, etc.
- cluster 4: topics related to databases, data, queries, database relations, etc.

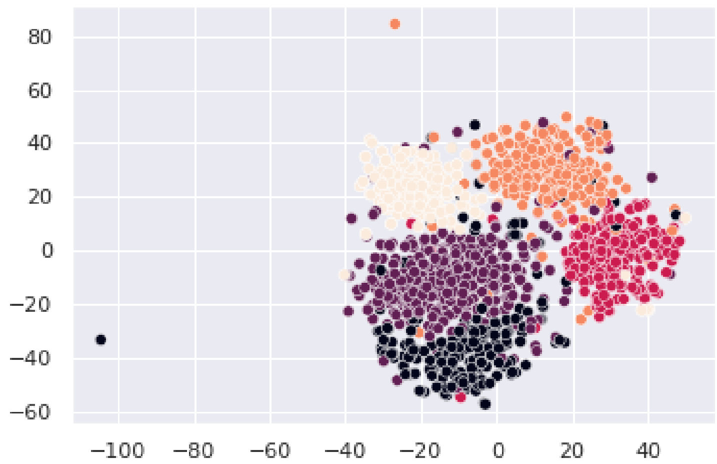
4 Instructions

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

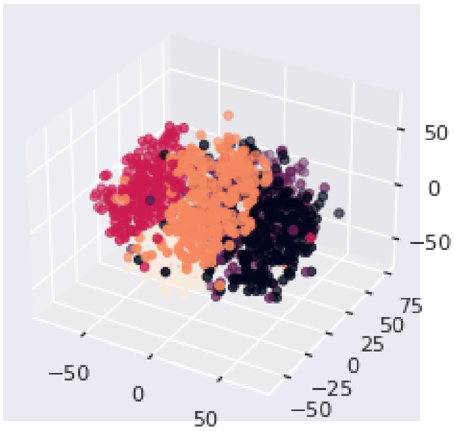
- `numpy` - used for numerical operations
- `pandas` - used to load and manipulate data
- `nltk` - used for stemming/lemmatization and stopword removal
- `sklearn` - used for performing tfidf vectorization, clustering as well as data dimensionality reduction
- `matplotlib` and `seaborn` - used for visualization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

Appendices

A Cluster Plots

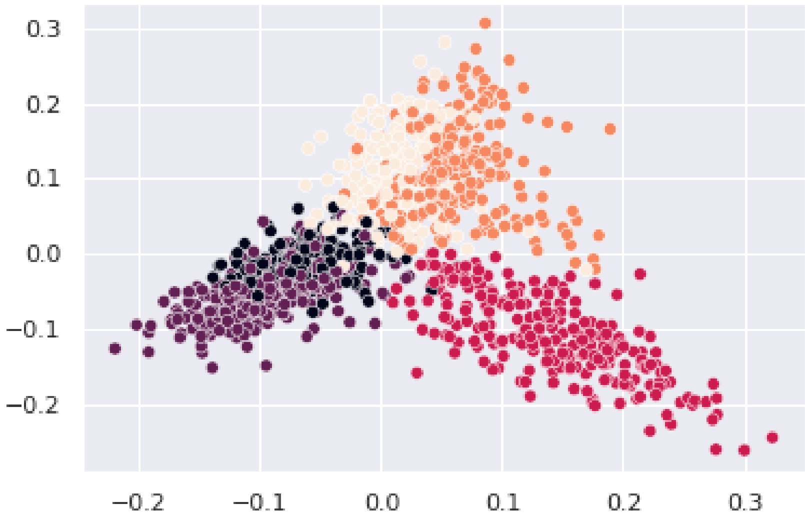


(a) 2D

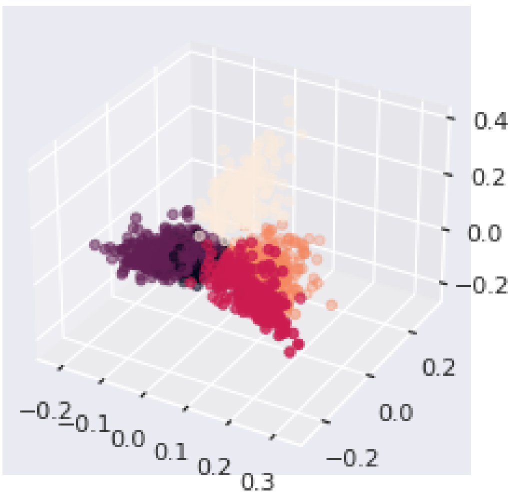


(b) 3D

Figure 1: Dimensionality reduction using t-sne.

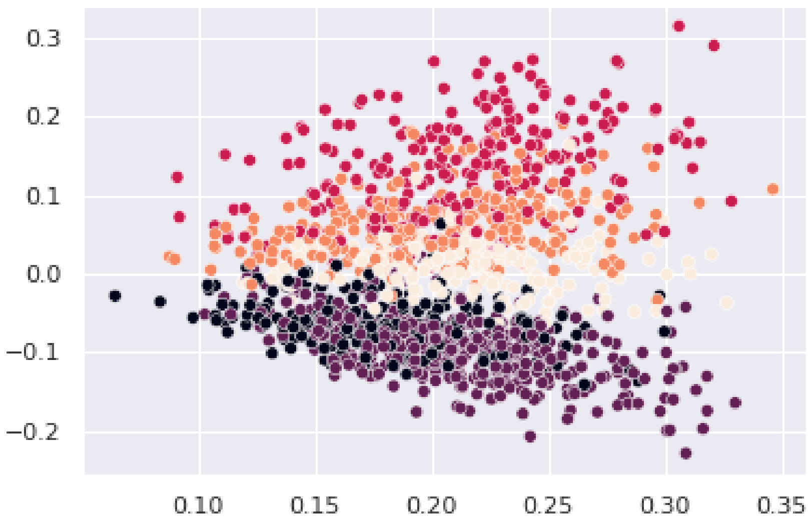


(a) 2D

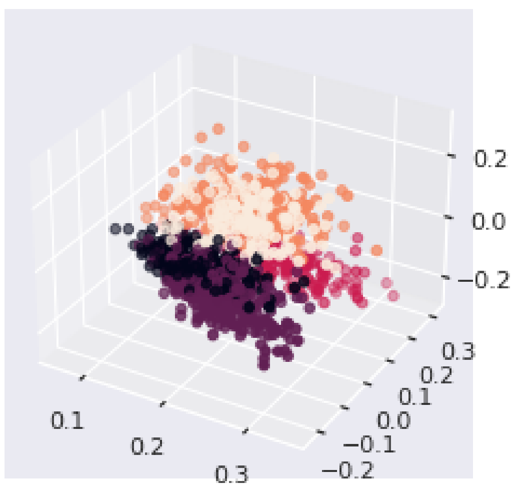


(b) 3D

Figure 2: Dimensionality reduction using PCA.



(a) 2D



(b) 3D

Figure 3: Dimensionality reduction using TruncatedSVD.

B Word Frequencies and WordClouds

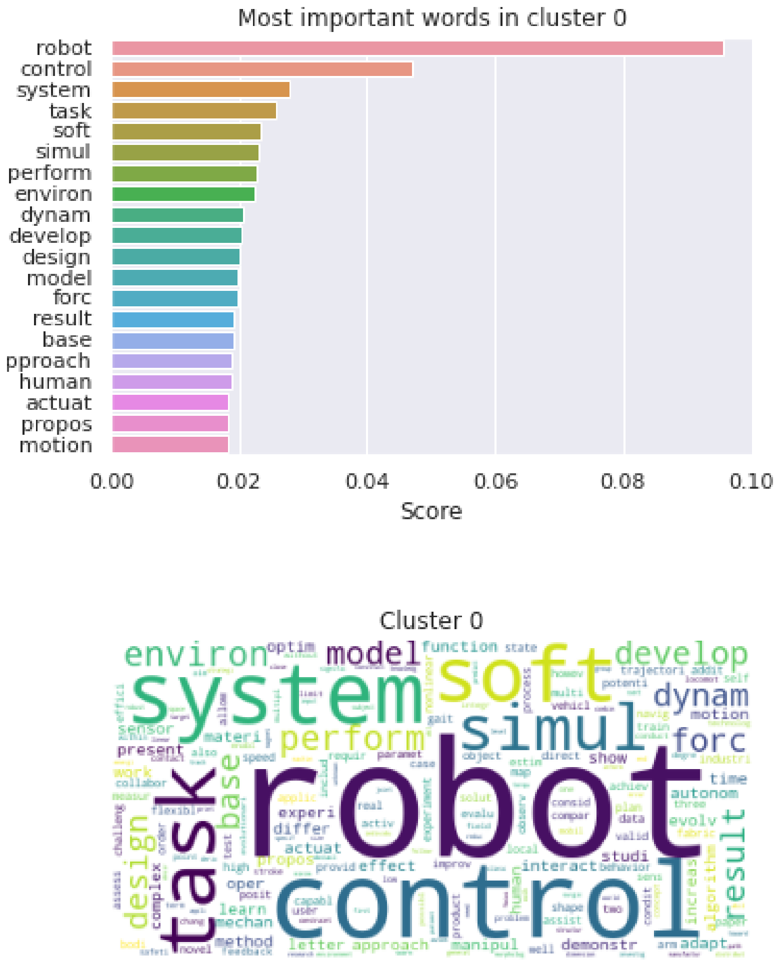


Figure 4: Cluster 0 most common words and the corresponding WordCloud.

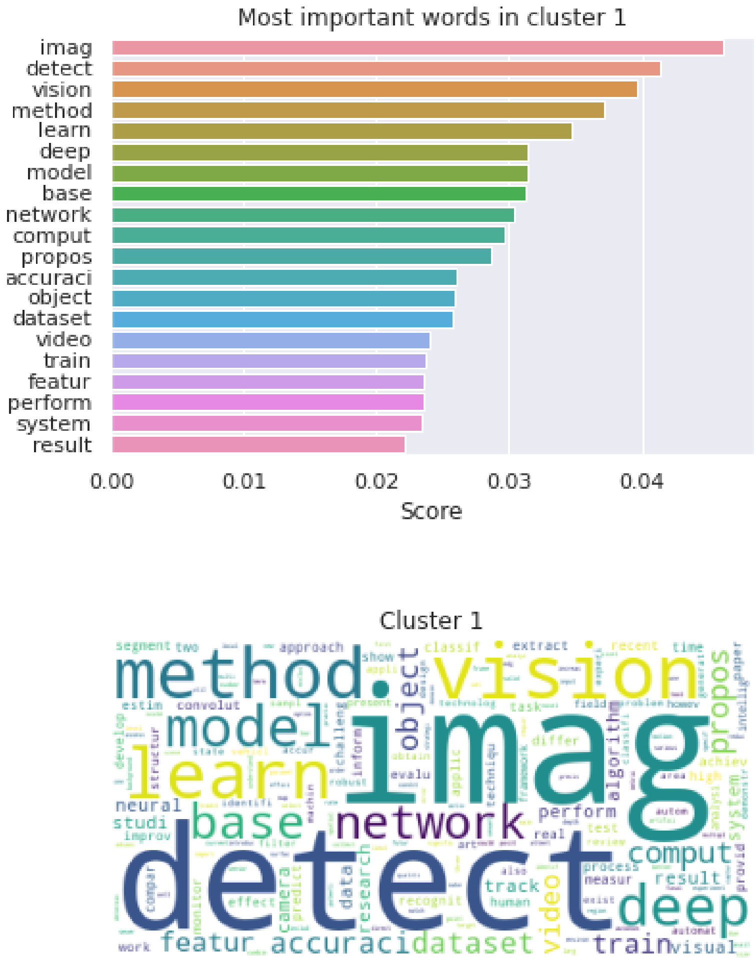


Figure 5: Cluster 1 most common words and the corresponding WordCloud.

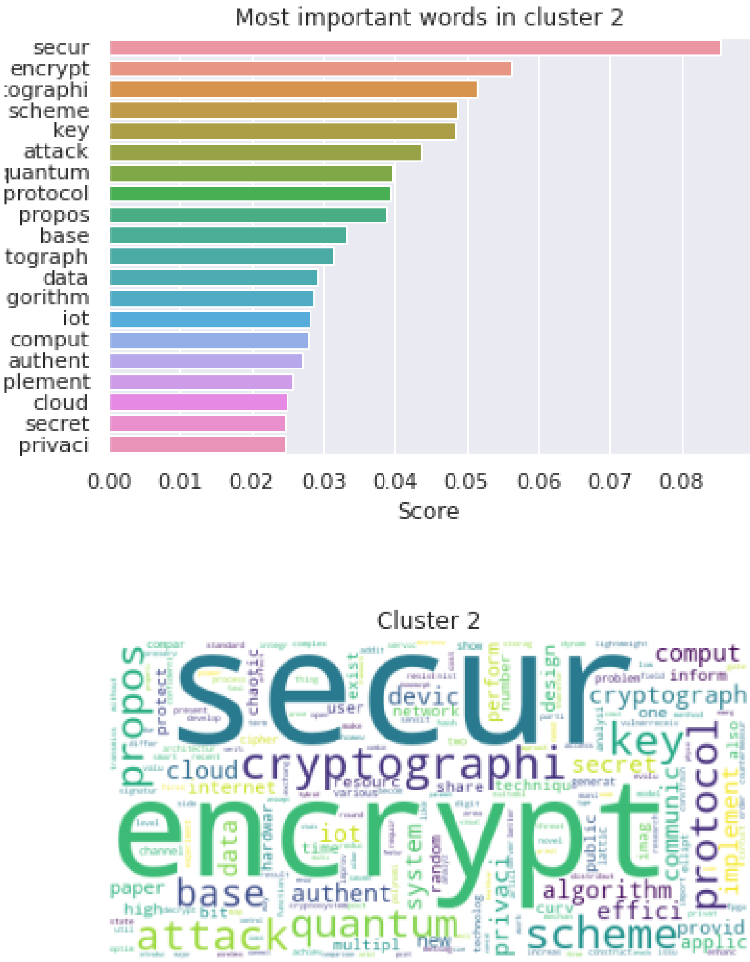


Figure 6: Cluster 2 most common words and the corresponding WordCloud.

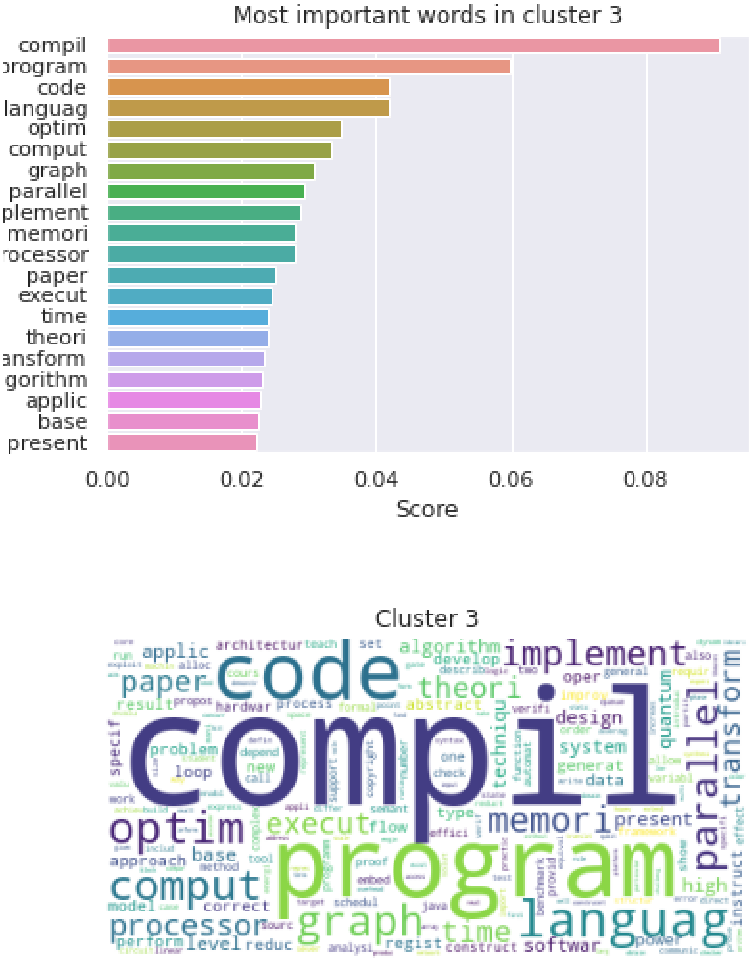


Figure 7: Cluster 3 most common words and the corresponding WordCloud.

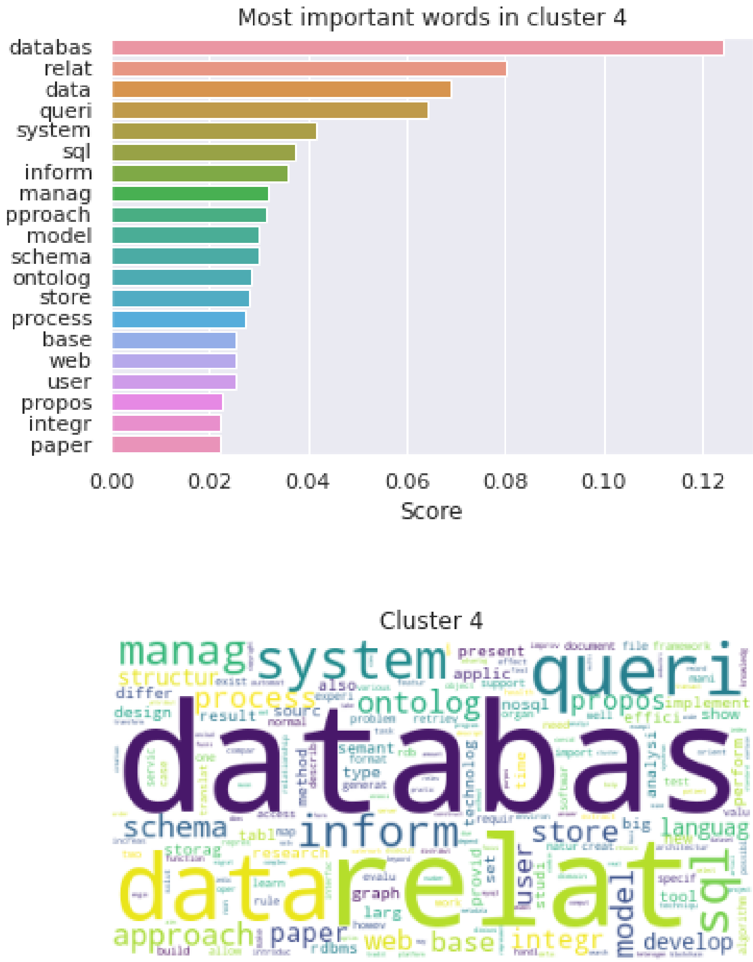


Figure 8: Cluster 4 most common words and the corresponding WordCloud.