Assignment 3 for APML 2022

Group 9
Christian Acosta | Josh Bleijenberg | Mischa van Ek | Sjoerd Vink | Robin Wiersma

Abstract

This paper examines various algorithms used for topic modelling and clustering of text documents, as part of the natural language processing paradigm. The studied documents are tweets about the corona pandemic regarding testing. Latent dirichlet allocation is used to perform topic modelling. Based on the perplexity score and human judgement, the ideal number of topics in the data set is 5. The resulting topic seem to have some overlap, which is possibly caused by the same overreaching theme (the corona pandemic). Agglomerative and KMeans is used to perform text clustering. The silhouette score of the Agglomerative clustering was 0.012, and of the KMeans was 0.020. The difference between these scores is marginal. However, the KMeans clustering algorithm seems to perform better on the data set based on human inspection.

1 Introduction

Natural language processing (NLP) is a widely used principle to process and analyze human language. It has a wide variety of applications, ranging from speech recognition to text classification, and from sentiment analysis to dialog systems. This study looks at topic modelling and text clustering. The methods are being applied to tweets about the corona pandemic regarding testing. The used data set is described in the second chapter of this paper. When specifically looking at implementation of these algorithms, it can for example be very useful for the Dutch government. Measures to prevent the spread of corona are a trade-off between the number of infections and social pressure. It would be incredibly helpful for the government to know what goes around on social media. By using topic modelling and clustering, the social pressure against which to weigh becomes measurable, making it easier to take informed decisions about the corona measures.

The main challenge in nlp is that text data is unstructured, making it hard to process and use. A righteous preprocessing pipeline is therefore of utmost importance before being able to apply topic modelling and clustering techniques. This, together with more information about the algorithms used is discussed in chapter three. The fourth chapter gives an overview of the performance of the different models, followed by an evaluation and a discussion. Finally, the conclusion of the study is discussed in the last chapter.

2 Data

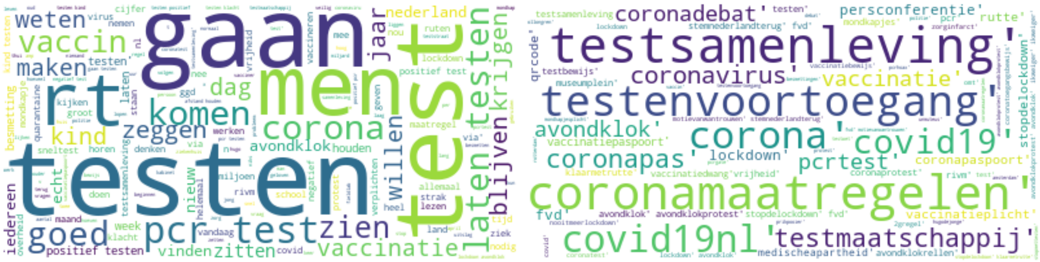
The data set provided consists of nominal data only. This nominal data contains about 100k raw tweets about the corona pandemic. These tweets contain only the words and have no further characteristics such as the time of the posted tweet or the original name of the poster. The tweets were collected based on 10 themes, ranging from lockdown to face masks. This article examines the tweets on the theme 'testing', e.g. PCR and antigen tests.

Understanding the most commonly used words is an important step in understanding the data set. Figure 1a shows a word cloud of the entire preprocessed data set. The data set comes from a text file and has been preprocessed with lemmetaization and tokenization. The following sub chapter 3.1

Preprint. Under review.

discusses the preprocessing of text data in more detail. This word cloud displays, ’testen’, ’test’ and ’gaan’ to occur most frequent as tokens coming from tweets related to corona pandemic. "a formal item whose pattern of occurrence can be described in terms of a uniquely ordered series of other lexical items occurring in its environment"[WK92]. For simplicity sake, we will refer token as words in this chapter. Surprisingly words as "vaccine" and "corona" rank lower as reoccurring words in tweets.

Hashtags are a critical part of tweets. A hashtag is a word or phrase preceded by a hash sign (#). Hashtags help identify tweets on a specific topic. Therefore a wordcloud was also made to show the most popular hashtags that occured in tweets. In figure 1b



(a) Wordcloud

(b) Hashcloud

Figure 1: Dataclouds

3 Methods

Several topic modelling and clustering algorithms have been implemented for natural language processing on the provided tweets about the corona pandemic. The algorithms used are derived from Spacy (version 3.2.1), SKLearn (version 1.0.1) and tweet-preprocessor (version 0.6.0). Also, a pre-trained Dutch language model from Spacy is used, called nl-core-news-sm. The first step in the process was the text data preprocessing, followed by the text data representation. Finally, the algorithms are implemented which cluster the text data and model the topics. An overview of the taken steps is shown in Figure 2.

3.1 Text data preprocessing

The saying ’garbage in, garbage out’ is especially applicable to natural language processing. Some word-types, for example stop words and punctuation’s, should not be taken into account in the topic modelling or clustering process. It influences the result in a bad way. In order to arrive at the best possible answer, it is important to preprocess the tweets.

Filtering The data set consists of tens of thousands of tweets, all with the topic of the corona pandemic. This article only uses the tweets with the subject of testing. The tweets were filtered by the following keywords: test, testing, tested, pcr and antigen. After filtering the tweets on these keywords, more than 12,000 tweets remained.

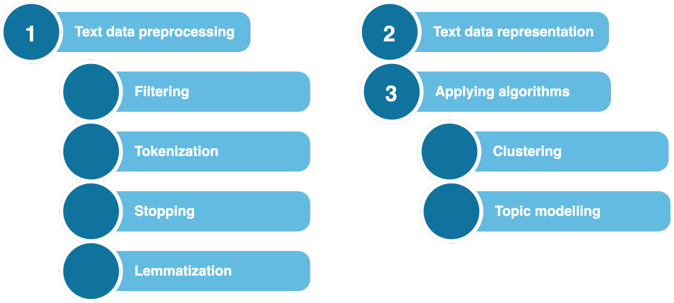


Figure 2: Preprocessing pipeline

Tokenization The next step is tokenizing the words in the tweets. Tokenization is the process of segmenting text data into words, called tokens. This is done by splitting the sentence at non-letter characters (spaces). The biggest challenge in this process is that tokenization is language specific. However, the Spacy library provides a pre-trained model for the Dutch language, making it easier.

Stopping The next step in the preprocessing pipeline is cleaning up the tokenized tweets by removing insignificant tokens. Examples of these are URLs, emoji's and reserved words. These words have no further meaning in natural language processing. It even influences the result in a bad way. This is because words like 'and', 'a' and 'the' are very common wouldn't discriminate between topics. In addition, the stop words and punctuation's can also be removed. This is done with the Spacy library and its pre-trained Dutch model.

Lemmatization The final step in the preprocessing pipeline is lemmatization. Similar tokens between tweets can have different forms, for example verbs and plurality. Consider two distinct tweets containing a similar word, only in a different form. Let's say that these words are 'cat' and 'cats'. In the text data representation, these words are considered to be different. However, this probably isn't the case because cat and cats are the same animals, only in different plurality. The solution for this is mapping the tokens to a common base form. Lemmatization is the process of handling these types of plurality and bringing back verbs to the stem. This is done with the Spacy library and its pre-trained Dutch model.

3.2 Text data representation

Before the preprocessed tweets can be used for clustering and topic modelling, it is important to represent the data in such a way that they can be used. They must be represented in a structured way in order for an algorithm to be able to use them. A common way for text data representation is a term-document matrix, also called a bag of words. The idea is relatively simple. A sparse matrix is created for all the words in the data set, where each column is a word en the rows are the documents. By representing the tweets this way, it is possible to compare tweets and perform calculations. The bag of words is created by using a count-vectorizer from the SKLearn library.

3.3 Algorithms

Natural language processing is the use of a computer program to understand human language. It has a wide variety of applications, ranging from speech processing to syntactical analysis. In this paper two specific applications are being examined, namely topic modelling and clustering. The final outputs of topic modeling and text clustering are very similar but use different approaches to get results.

3.3.1 Topic modelling

Topic modeling uses a statistical model to discover topics in a collection of documents. The goal here is to identify a list of topics, where each topic is a cluster of words that frequently occur together.

Latent dirichlet allocation (LDA) is one of the most widely used statistical models in topic modeling. It builds on latent semantic analysis (LSA), which is used to find a low-dimensional representation of documents and words. LDA is a probabilistic model that can be used to explain the observed variables (tweets) by hidden variables (topics). The model assumes that tweets from the same topic contain a similar collection of words. It takes as input a bag of words and a number of topics. The output is a probability distribution for the words belonging to a certain topic and the topic distribution of all given documents.

The main challenge with LDA models is that the returned topics are soft-clusters. There is no binary answer, meaning that it is about grouping tweets such that a tweet can be part of multiple clusters. This makes it hard to decide whether the parameter (number of topics) is correct. The evaluation is also tricky case. There are quantitative metrics, but those are poor indicators for measuring model performance. The evaluation process is extensively discussed in Chapter 4.1.

3.3.2 Clustering

Clustering groups documents into different clusters based on similarity. Each document is represented by a vector representing the weights assigned to words in the document. The goal of clustering is grouping tweets such that tweets within the same cluster shall be as similar as possible and tweets of different clusters shall be as dissimilar as possible. Two clustering algorithms have been examined in this paper.

Agglomerative clustering The first clustering algorithm is Agglomerative clustering. Agglomerative is an unsupervised learning model that computes clusters from bottom-up. This means that is first calculates the distance between all individual tweets, and merges the closest ones together. Because it computes all the distances, the main disadvantage of this algorithm is that it isn't very efficient. It should also be taken into account that it cannot handle outliers very well. Outliers can quickly become its own cluster, making the important clusters merge together. The main advantage is that there is no need to define the numbers of clusters preliminary.

KMeans clustering The second clustering algorithm is the KMeans algorithm. KMeans is a partitioning based clustering method. This means that it divides the data set in partitions, regardless of density. It defines the clusters by putting k points arbitrarily in the data set. With each iteration, the algorithm moves the points to the mean of the closest tweets around it. A tweet belongs to the cluster it is closest to.

It is a very efficient algorithm that is able to handle a large amount of documents. The main drawback of this algorithm is that the number of clusters needs to be specified preliminary. The solution for this is to base the number of clusters in a quantitative measure. This is further discussed in Chapter 4.2.

4 Evaluation and Discussion

The evaluation of unsupervised learning algorithms isn't as straightforward as with supervised learning. Natural language is ambiguous and full of subjective interpretation. There isn't one single version of truth, making it harder to evaluate these types of algorithms.

4.1 Evaluation metrics

Especially with topic modelling, which topic an article belongs to is open for interpretation. The preferable and most reliable way is to incorporate human judgement in the evaluation process. In order to do this, it is necessary to make some sort of visualization. The visualization result is discussed in the next paragraph. It is also possible to use a more quantitative metric when measuring performance. The perplexity score is often used for how well a probability distribution predicts a tweet. The higher the perplexity score, the harder the topic modelling problem is.

Using human judgement is also one of the best methods to evaluate clustering. The visualization result is discussed in the next paragraph. It is also possible using a more quantitative approach in the evaluation process here. There are several metrics to evaluate performance. The most used one is the silhouette coefficient. The final score ranges between -1 and 1. -1 means that there aren't any good clusters, 0 means that the clusters overlap and 1 means there are good clusters. The silhouette score is calculated by comparing the intra-cluster distance (distance inside the clusters) and the inter-cluster distance (distance between clusters).

4.2 Overview of model performance

4.2.1 Topic modelling

To visualize the results of the LDA topic modelling algorithm in an interactive way, a dashboard was created using the pyLDAvis Python library. This method allows one to interpret the topics in the LDA model fitted to a corpus of textual data. Figure in Appendix C shows a sample view of the dashboard. Figure 3 shows the most common words that occur together in a topic.

The perplexity score of the model with 5 topics is 3261. When increasing the number of topics, the perplexity score increases significantly. However, when decreasing the number of topics, the

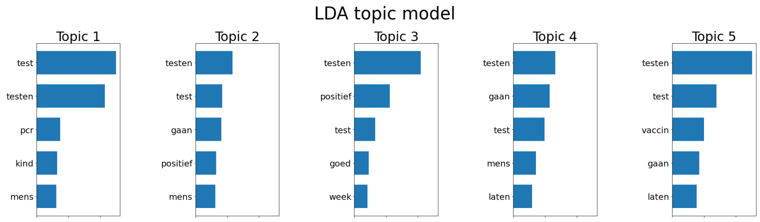


Figure 3: LDA topics

perplexity score decreases only slightly. Because of this, 5 topics is the ideal trade-off between perplexity and the number of topics.

4.2.2 Clustering

Agglomerative The first clustering algorithm is the Agglomerative clustering, based on hierarchy. Based on human inspection, 4 clusters was the best option as a parameter. Figure 4 shows the most common words that occur together in a cluster. Looking at these words, it is quite hard to distinguish clusters from each other. There seem to be a lot of similar words, or synonyms. However, when looking at the scatter-plot visualisation in Appendix C at Figure 8, clusters are clearly visible. The silhouette score of the model is 0.012, meaning that there is some overlap between clusters. The visualisation in Appendix C at Figure 8 confirms that.

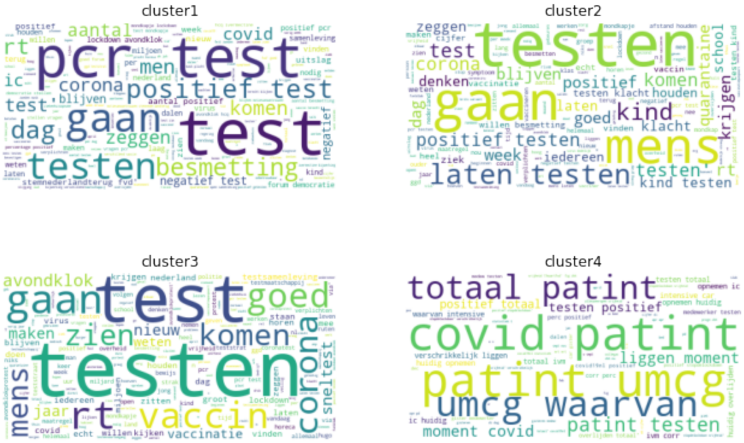


Figure 4: Agglomerative clustering

KMeans The second clustering algorithm used is the KMeans algorithm. The hardest challenge is choosing the k preliminary. A method to do this is the elbow method. When calculating the inertia, it can be seen that an 'elbow' emerges at k=4. Meaning that 4 is the optimal k for this document set when looking at the inertia. The inertia graph is shown in Appendix A at Figure 6.

Figure 5 shows the most important words per cluster in a word cloud. When looking at these words, there seems to be a specific theme in each cluster. For example, cluster 2 contains the words child, parent, school, complaint and quarantine. These tweets will probably be about the situation going on around schools and infections at school.

For visualization purposes, a range of 4 different K's were chosen to show the impact on a scatter-plot. The calculated silhouette score for the model with k=4 is 0.020, which means that there is a slight overlap between the clusters. It is striking that the silhouette score of models with the other k's is slightly higher. However, this difference is negligible. The scatter-plots of the models with different k's are shown in Appendix A at Figure 7.

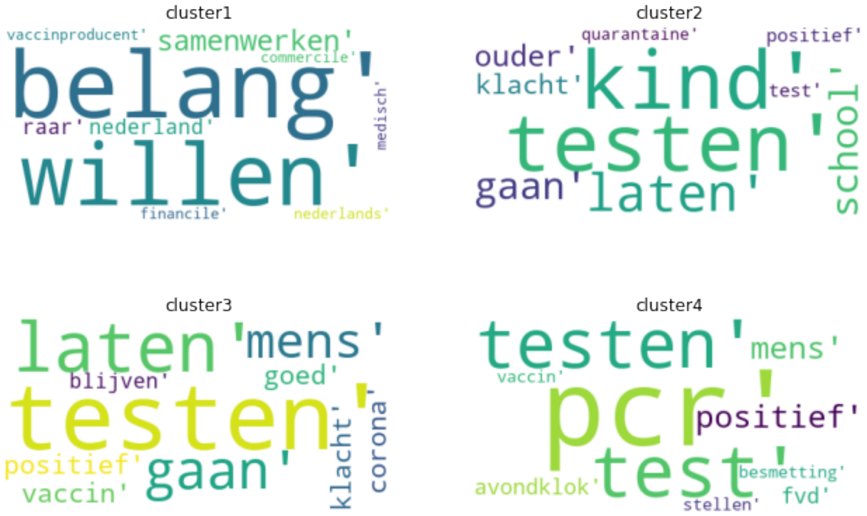


Figure 5: KMeans wordcloud

5 Conclusions

To summarize, the goal of this assignment was modelling the different topics and clustering similar tweets of a document set derived from Twitter.

The LDA model results in 5 topics, each with a frequent word set. There is some overlap between topics. This can be explained by the relatively high perplexity score, which indicates that it is hard to cluster the tweets. A possible cause of this can be that the topics of the tweets all have the same overreaching topic (the corona pandemic).

The results of the clustering algorithms turned out to be very similar. The difference between silhouette scores was negligible. However, when looking at the word clouds generated by the models, the KMeans algorithm seems to be the better option. The clusters revolve more around distinct topics, unlike with agglomerative clustering. There has been experimented with density based clustering (DBSCAN) as well, only the results were not very well. A possible explanation for this is that the tweets are very similar to each other. This makes it probably hard to distinguish areas with high densities. The result of this was one very big cluster.

5.1 Future work

This current paper studied both topic modelling and text clustering, resulting in a quantitative analysis about the topics and clusters of tweets. Possible future research can also carry out sentiment analysis. This will analyse the sentiment of tweets, giving a more in depth view and making it even more useful for the Dutch government.

6 Contributions of Group Members

6.1 Contributions of Group Members

Assignment 3 - Natural Language Processing	**Team members**
Task 1: Exploring the data set	
1.1 Data Processing	Sjoerd Vink
1.2 Exploratory Data Analysis	Mischa van Ek
Task 2: Outcome	
2.1 Topic Modelling	Robin Wiersma
2.2 Results, evaluation and Interpretation	Christian Acosta
Bonus Tasks	Josh Bleijenberg

Report	**Team members**
Abstract	Sjoerd Vink
1. Introduction	Robin Wiersma
2. Data	Mischa van Ek
3. Methods	Sjoerd Vink
4. Overview and comparison of algorithms	Christian Acosta
5. Conclusion	Sjoerd Vink

6.2 Group reflection

Looking back at the advancement of the assignment, we can conclude that it was more challenging than the previous assignments. None of us had yet any experience with natural language processing, making it harder to complete the tasks. The demonstration of the teaching assistant helped a lot with our progress. The collaboration between team members went well. Also, the assignment overlapped the Christmas break which brought some planning challenges. However, these challenges have ensured that we even spend more time on the assignment. As a result, we are very satisfied with the end product.

References

[WK92] Jonathan J Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*, 1992.

Appendices

A KMeans visualizations

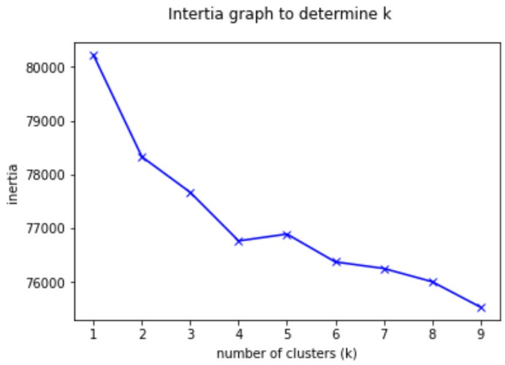


Figure 6: Inertia graph of KMeans algorithm

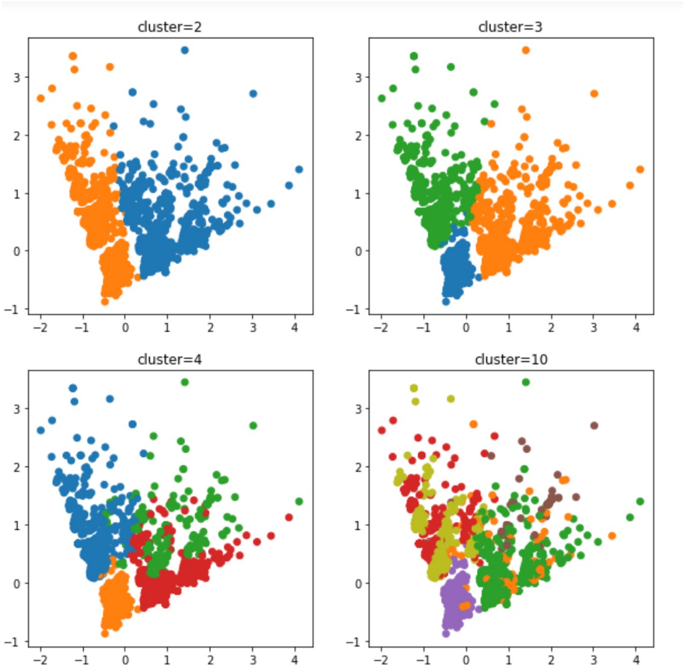


Figure 7: Scatter plots of different K's

B Agglomerative clustering

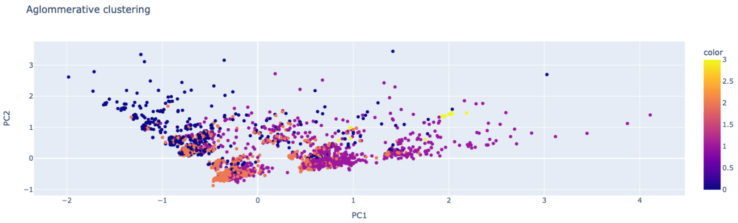
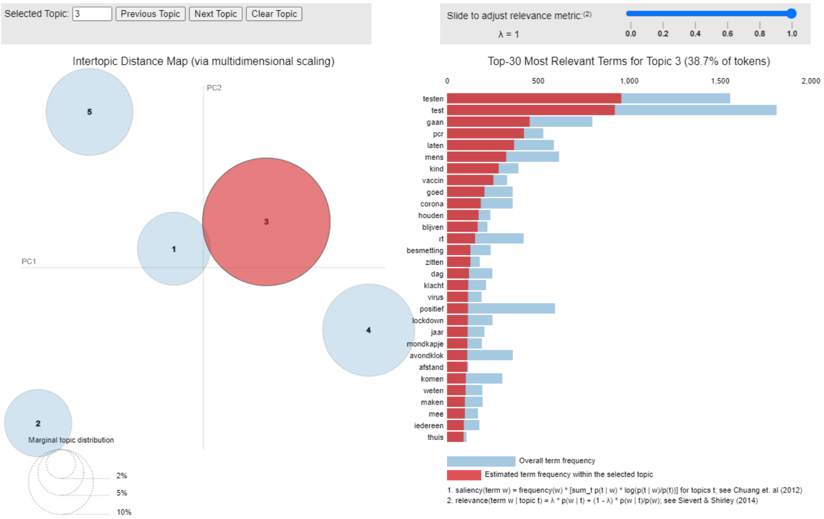


Figure 8: Agglomerative clustering

C Topic modelling dashboard



Selected Topic: 3

Slide to adjust relevance metric:(2)

λ = 1

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 9: LDA dashboard using pyLDAvis