INFOB3APML: Assignment 3

Group 2

Lorenzo Marogna Minna Vainikainen Tomás Tavares

January 2024

Abstract

In this comprehensive analysis of topic identification in social media data, this report investigates three unsupervised algorithms for natural language processing. Specifically, one topic modeling algorithm called LDA, and two clustering models called k-mean and DBSCAN. The raw text data provided from Twitter will be preprocessed with the python spaCy library providing tokenization, stopping, stemming and lemmatization. The outcome will be a set of data to which the algorithms can be applied. Finally, the report concludes with an evaluation and comparison of the different models by using visualization methods. Additionally a further comparison of the clustering results is performed by using some objective metrics such as silhouette score, Calinski Harabasz score and Davies bouldin score. The findings indicates that LDA consistently outperformed the clustering methods in terms of topic coherence and interpretability. When comparing the clustering methods k-means demonstrates better-defined and compact clusters.

1 Introduction

There are different strategies for addressing topic identification. This report will present two of them; topic modeling and text clustering. The goal of topic modeling is to discover topics that are frequently discussed in given documents. The goal of text clustering, on the other hand, is to group documents into different clusters based on similarity where each document is represented by a vector representing the weights assigned to words in the document. The purpose of this assignment is to identify topics in online social media data by comparing the performance of a topic modeling algorithm and a clustering algorithm. The data on which these algorithms will be applied comes from messages posted on Twitter related to the COVID-19 pandemic. The data will be filtered according to the theme loneliness.

2 Data

In order to identify topics in real social media data we will use text data of about 100K Twitter messages related to the COVID-19 crisis. Each message is in Dutch and has been published on the social media platform Twitter between January and November 2021. The dataset was produced by a random selection of raw twitter data where these messages were collected based on 10 themes, such as lockdown, face mask and social distancing. This assignment focuses on loneliness, which is one of these themes. The data will be filtered by loneliness **before** topic identification.

One difference between Twitter data and usual text data is that social media messages are not formally written and can often contain misspelled abbreviations and slang. These messages also include characters like emojis and URLs. When preprocessing the data, this causes a challenge.

In order to make an exploratory data analysis two word-cloud-plots are created. Figure 1 presents the original text data, while figure 2 presents the processed text data. A word cloud visualizes the frequency of words, i.e. when a word appears more frequently it becomes larger in the cloud. We can see that the most frequent words before processing the text data were coordinating conjunctions like “the”, “and”, “only”, “its”, and “but”. We can see a difference in which words are represented after they have been pre-processed. These words are more describing; “go”, “one”, “lonely”, “feel”, “really”.

Visualization techniques will be used in order to evaluate and interpret the results from the two methods. We use the python library pyLDAvis to create the interactive topic visualization. Word cloud and word count of dominant topics will be used to visualize clustering.

Figure 1: Word cloud based on original text data

Figure 2: Word cloud based on processed text data

3 Method and Experimental Setup

3.1 Data Pre-Processing & Feature Selection

The goal of the preprocessing steps is to present the text data as "a bag of words". The basic unit of text data is a word and, by presenting the data in this way, the grammar and ordering relations between the words are ignored. At the end of preprocessing we want to represent the text data points as vectors with numeric values. The preprocessing steps include tokenization, stopping, stemming and lemmatization.[1] In order to achieve this, we have chosen the python library spaCy which offers models for this.

When developing the code and testing the algorithms, the nlp model "nl core news sm" was used.This is a small Dutch pipeline optimized for CPU and works well for written text from news and media. The model has high accuracy and offers a lot of useful components for complex data.[2] When finally presenting the code and evaluating the results "nl core news lg" is going to be used. This is a model provides higher accuracy and better understanding of the language.

Tokenization is about segmenting a given text term into basic units, which basically means that the text is being split into words (tokens). The text data were filtered based on the theme of loneliness. Some topic words were set up; "geïsoleerd", "isolatie", "kluizenaar", "eenzaamheid", "eenzaam", "virtueel", "knuffel" and "thuis". These words were then used to create a data filter that filtered out 10384 messages related to loneliness. Initially, the raw text data contains URLs, mentions, numbers, hashtags and emojis, since it originates from messages on social media. When preprocessing the data, we got rid of these characters that cannot constitute a token. The processed messages were then added in a way that prevents duplicate messages and it resulted in 8416 unique messages related to loneliness.

We then had to convey the data into a format that can be used in clustering and LDA algorithms. There is

[1] Lecture 10 - Natural language processing
[2] https://spacy.io/models/nl

a problem that words can be used in different forms. Stemming and lemmatization are methods that manage this, by mapping those words to a common base form. Examples of what it can handle is capitalization of words, plurality of words, verbs in different tenses and words from the same root and with similar meanings.

3.2 Model Selection

Since the starting point of this assignment is a raw text data file, we have to acknowledge that we are dealing with an unsupervised learning problem. The collected twitter messages are in other words not assigned into classes by a human expert. In order to identify topics in the raw data file containing tweets, we have chosen to use two types of unsupervised learning algorithms.

The first method is topic modeling and it is used to automatically discover patterns in the unsorted data. In this case, the definition of a topic is a group of words that are likely to appear in the same context. What a topic model does is to automatically group related words into cohesive topics and associate terms and documents with those topics. The Latent Dirichlet Allocation (LDA) algorithm is used for this purpose, which is one of the most popular models used for topic modeling.

The second method is clustering, which is the most common form of unsupervised Learning. [3] What a cluster algorithm does is to group a set of documents into subsets, what we call clusters. The goal of the algorithm is to create clusters that are coherent internally, but different from each other. Distance metric is a crucial element in a clustering algorithm, as it describes the way distances are measured, thus directly affecting the outcome. K-means is the clustering algorithm that will be used in this assignment because it is characterized by two important distinctions. Firstly K-means is a flat clustering algorithm, which means that it creates a structure that does not relate clusters to each other. This is an advantage in this case as we are simply not interested in how tweets belonging to different clusters are related to each other. Secondly, K-means computes a hard assignment, which means that each element belongs to only one cluster. This is also an advantage as the task is to investigate a specific theme.

The vector representing the sentence is computed as the average vector of all the vectors representing words in the sentence. The aim is to summarize the whole vector as a single, unique word with a fixed length.

3.2.1 Latent Dirichlet Allocation (LDA) algorithm

Latent Dirichlet Allocation algorithm takes a term-document matrix, concentration parameters and the number of topics as the input. The output is given by the word distribution of topics and the topic distribution of all given documents.

As with all models, there are also weaknesses in using LDA as a topic model. Firstly, LDA ignores the order of words, which could result in a source of error where the message is misinterpreted. We also need to specify the dimension of the latent space, i.e. number of topics. We thus seek a value for this, large enough to capture the true structure of the given text data, but also small enough to exclude sampling errors or unimportant information. There is no accurate way to set up this value in advance and therefore we need to try different values and compare the performance. [4]

For the sake of this study, LDA was requested to build 5 topics, while the rest of the parameters were left to the default value.

3.2.2 K-mean clustering

The particular objective function that K-means is optimizing is the sum of squared distance from any data points to its assigned center. This is a natural generalization of the definition of a mean. The first step of K-mean is to randomly select an initial cluster center. By iterations, the algorithm then moves the cluster centers around in order to minimize the objective function.

The key problem with unsupervised learning is that we have no way of knowing what the "right answer" is. Convergence to a bad solution is usually due to poor initialization, even so in this case. Initialization is the biggest practical issue in K-means as poorly initialized cluster means often result in convergence to uninteresting solutions.

The algorithm implementation is the one provided by sklearn, with the downside of not allowing the user to use the cosine similarity as a metric of distances (default one is euclidian). KMeans is based on the minimization of the sum of squared Euclidean distances between data points and cluster centers. The algorithm involves updating cluster assignments and cluster centers based on this Euclidean distance metric. Changing the distance metric to something like cosine similarity would require a different approach to the optimization problem and may not fit well with the underlying assumptions of the KMeans algorithm. [5]

[3] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008

[4] Lecture 12 - Topic modeling

[5] Hal Daumé III, A Course in Machine Learning, 2017

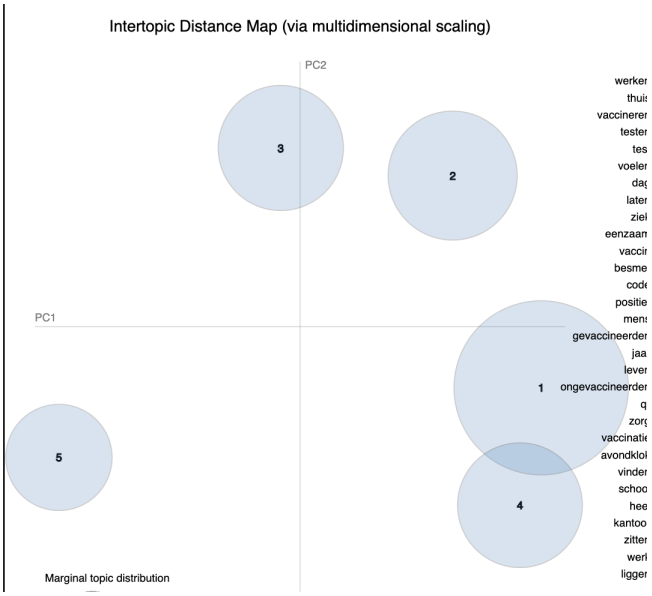

Figure 3: Latent Dirichlet Evaluation - PyLDAvis

Consistently to 4.0.1, K_means was set to produce 5 clusters.

3.2.3 DBSCAN

DBSCAN, Density-Based Spatial Clustering of Applications with Noise, is a density-based clustering algorithm. This algorithm finds core samples of high density and expands clusters from them and works good for data which contains clusters of similar density. Two parameters are selected; min samples and eps. Min samples is the number of samples in a neighborhood for a point to be considered as a core point. DBSCAN will find denser clusters if min samples is set to a high value and more sparse clusters if it is set to a lower value. We set min samples to 6. Eps is the maximum distance between two samples for one to be considered as in the neighborhood of the other. This is not a maximum bound on the distances of points within a cluster and is set to 1.3. As the alghorithm is very sensible to these parameters, they were set in order to produce an acceptable amount of clusters. The advantage of using DBSCAN compared to Kmeans relies in the possibility to use cosine similarity as a metric, which might be a better approach for our usecase.[6]

4 Evaluation & Discussion

In this section, we will make some considerations of how the topics created with methods from section 3.2 are well identifying a specific and unique semantic area compared to other topics. For the first part, considerations will be based on observation and visualizations, which are intrinsically corrupted by human judgement. Sequentially, we'll try to give an objective evaluation using some popular methods for clustering evaluation.

4.0.1 Latent Dirichlet Allocation (LDA) algorithm

The evaluation for LDA results is done with the help of PyLDAvis package, providing a function for interactive visualizations. As we can see in figure 3, a principal component analysis shows that topics are rarely overlapping (at least considering PA1 and PA2) and proportionate in size.

Most salient terms for topic 1 are 'feel', 'going', 'home', 'man', and 'lonely' (Figure 4). As topic 1 is the most relevant in size, it is a positive result to see how it represents the theme for which messages were filtered (loneliness). It seems from the results that other topic are trying to cover different nuances of the theme. For example, topic 5 salient terms are 'vaccins', 'man' and 'leave' (Figure 5).

4.0.2 K-mean clustering

The outcome of a clustering algorithm is a series of labels to apply to each document, so that similar documents get to be assigned to the same cluster. After identifying similar documents and grouping them, the group is summarized as a bag of words. This BoW is builded by analyzing the words with the highest frequency among the group.

[6]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

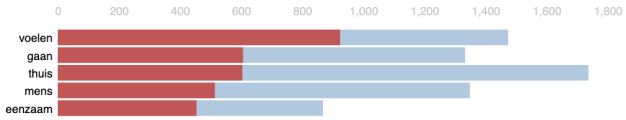

Figure 4: Most salient terms for Topic 1

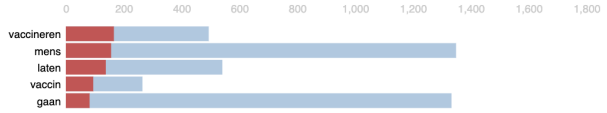

Figure 5: Most salient terms for Topic 5

Figure 6: Word Clouds for topic modeling with k means clustering

Some limitations are evident while looking at the result. In particular, topics from different clusters share the majority of popular words. This might be the effect of filtering messages with the same theme, ending up with a series of similar messages. Moreover, Kmeans algorithm implementation is using euclidean distance, where the size of vectors matters. It might be interesting to implement a variation of Kmeans called spherical k-means, which uses cosine similarity as a metric and normalized vectors.

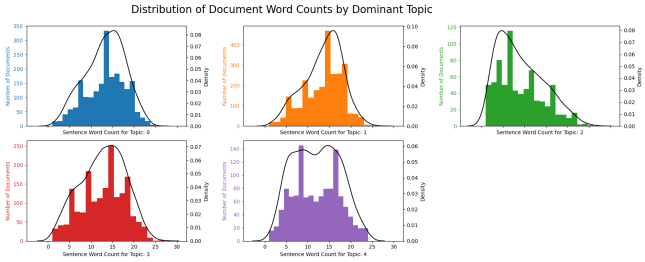

Figure 7: Distribution of words with k means clustering

Accordingly to figure 7, documents of different clusters tend to have the same length (approximately 15 words). This is not surprising, as the document vector is produced by averaging all the word's vectors in the document, independently of their number.

4.0.3 DBSCAN

As we can see in table 1, DBSCAN individuated a main cluster 0 and some smaller clusters. This is the best we could do by varying eps and min_samples in order to have multiple clusters. The big downside of this approach is that a low-density space led to the majority of the elements to be classified as noise.

cluster	# elements
-1	5746
0	2608
1	15
2	11
3	9
4	6
5	7

Table 1: Clusters' label and size

Since almost all documents were considered as belonging to the same, big cluster, topic 0 is the most representative of the data. Again, this could be the consequence of filtering messages regarding loneliness in advance.

The BoW for each cluster is computed at the same way as 4.0.2.

Figure 8: Word Clouds for topic modeling with DBSCAN clustering

4.1 Performance Comparison

4.1.1 LDA and Clustering-base method

Upon evaluating the performance of LDA and the clustering-based approach, it was observed that LDA consistently outperformed the clustering methods in terms of topic coherence and interpretability.

Latent Dirichlet Allocation demonstrated superior performance in extracting coherent and semantically meaningful topics from the dataset. The coherence scores across various values of k indicated that LDA consistently produced more interpretable topics.

Multiple factors contribute to the superior performance of LDA: firstly, being a probabilistic generative model that explicitly models the probability distribution of words in topics allows for a more nuanced understanding of document-topic relationships compared to the clustering approach. Moreover, LDA leverages word co-occurrence patterns within documents to capture contextual information.

4.1.2 Kmeans & DBSCAN - evaluation metrics

As the human judgement of a cloud of words is essential but also subjective (and our comprehension of dutch is fairly limited), some popular metrics for clustering evaluation were adopted. In particular, we used silhouette score, Calinski Harabasz score and Davies bouldin score.

- The Silhouette score measures how well-separated clusters are, with a higher silhouette score indicating better-defined clusters. The silhouette score for a single data point i is calculated using the following formula:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where:

  - $S(i)$ is the silhouette score for data point i.
  - $a(i)$ is the average distance from the i-th data point to other data points in the same cluster (intra-cluster distance).

- $b(i)$ is the smallest average distance from the i-th data point to data points in a different cluster (inter-cluster distance).

The overall silhouette score for the entire clustering is the average of the silhouette scores for all data points.

In these formulas, distances was measured using cosine distance, which should be an advantage for DBSCAN which was trained with the same metric. The silhouette score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A score near 0 indicates overlapping clusters.

- The Calinski-Harabasz score is calculated using the formula:

$$\text{Calinski}(K) = \frac{B(K)}{W(K)} \times \frac{N-K}{K-1}$$

where:

- K is the number of clusters.
- N is the total number of data points.
- $B(K)$ is the between-cluster variance.
- $W(K)$ is the within-cluster variance.

The Calinski-Harabasz Index evaluates the quality of a clustering by measuring the ratio of between-cluster variance to within-cluster variance, with higher values indicating more well-defined clusters. It considers how compact clusters are while also taking into account the separation between them.

- The Davies-Bouldin Index is calculated for a clustering with K clusters using the formula:

$$DB = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{S_i + S_j}{d(\mu_i, \mu_j)} \right)$$

where:

- K is the number of clusters.
- S_i is the average distance from the centroid of cluster i to its points.
- $d(\mu_i, \mu_j)$ is the distance between the centroids of clusters i and j.

DB index aims to find clusters that are well-separated from each other and have high intra-cluster similarity. A lower Davies-Bouldin Index is indicative of a better clustering result.

Model	Silhouette score	Calinski Harabasz score	Davies Bouldin score
Kmeans	0.06	404	3.23
DBSCAN	-0.11	35.3	2.91

Table 2: Evaluation metrics

As we can see in table 2, objective metrics confirms the poor results of the clustering methods, regardless of the type of distance used by both the alghorithm and the score.

In K-Means clustering, the Silhouette Score (0.06) suggests some cluster overlap, while the Calinski Harabasz Score (404.0444) indicates well-defined, compact clusters. The Davies Bouldin Score (3.2395) is moderate, with potential for improvement.

For DBSCAN clustering, the negative Silhouette Score (-0.1117) indicates issues with cluster assignments, and the Calinski Harabasz Score (35.2542) suggests less distinct clusters. However, the Davies Bouldin Score (2.9183) is lower than K-Means, indicating better compactness and separation.

The clustering scores suggest that while K-Means demonstrates better-defined and compact clusters, DBSCAN faces challenges with cluster assignments and distinctiveness, requiring further exploration and parameter tuning.

5 Conclusion

The results show that LDA consistently outperforming when comparing it to the clustering-based methods. This is demonstrated both in terms of topic coherence and interpretability. LDA allows a more nuanced understanding of document-topic relationships compared to the clustering approach, by being a probabilistic generative model that explicitly models the probability distribution of words in topics and also captures contextual information for words. When comparing the clustering methods K-Means demonstrates better-defined and compact clusters, while DBSCAN faces challenges with cluster assignments and distinctivenes. Based on the comparison of the cluster scores, the general conclusion can be drawn that the two clustering methods with their chosen parameters gave generally poor results.

Group Members & Contributions

- **Lorenzo Marogna:**
  - Code: Task 2 & Bonus Task
  - Report: Evaluation & Discussion 4
- **Tomás Tavares:**
  - Code: Task 1
- **Minna Vainikainen:**
  - Report