Topic Clustering to Hot Event Detection

Pham Thi Bich Nga[1] and Dam Vu Trong Tai[2]

[1]Department of Computer Science, University of Information Technology
[2]Department of Computer Science, University of Information Technology

January 30, 2023

Abstract

Within the scope of the final project report for the Natural Language Processing course on the topic of Topic Clustering and using Topic Clustering for Hot Event Detection, we will propose a pipeline for Hot event detection based on Topic Clustering problem, experiment and evaluate with different clustering algorithms and improve them to find the most accurate pipeline. Then, use the evaluated model to apply to the Hot Event Detection.

1 Introduction

Nowadays, with a large amount of information published every day, it is very difficult to classify and identify important and core events by human effort. Therefore, Hot Event Detection problem has developed to meet the demand for condensed information. Hot event detection is a problem in information retrieval and event detection that aims to identify important or relevant events from large amounts of data in near real-time. This is often used to monitor and analyze social media, news articles, or other sources of information to identify emerging trends, topics, or events of interest to a particular audience.

Hot Event Detection based on Topic Clustering problem. Topic Clustering is a technique that groups similar documents or text items into clusters, based on the common themes or topics they cover. Topic Clustering is used to cluster data to determine which clusters are most likely hot events. We experiment with different Clustering Algorithms and improve them to use to choose the best pipeline.

This project consists of two main parts. The first part is proposing a pipeline to solve the Topic clustering problem, evaluating it on available public dataset from Scikit Learn, comparing and evaluating clustering algorithms. The second part is applying the pipeline with the best and most effective clustering algorithm to solve the Hot Event Detection problem, with the data being news crawled from two major magazines: The New York Times news and BBC news.

2 Topic Clustering

2.1 Dataset

The dataset we used to evaluate Topic Clustering model is The 20 Newsgroups Text Dataset from Scikit-learn. The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics (Target names) split in two subsets: one for training (or development) and the other

one for testing (or for performance evaluation). But in the context of this project, we only used the testing set of this dataset to evaluate the performance of our model.

Item	Quantity
Text docs	7532
Target names	20

Table 1: Statistics based on test set.

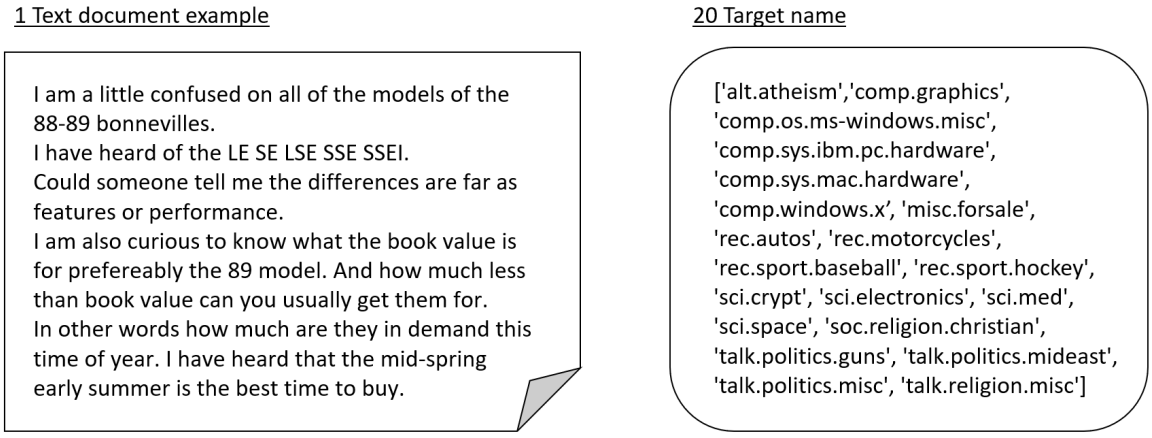

Figure 1: One example of text documents and 20 group target names.

2.2 Model

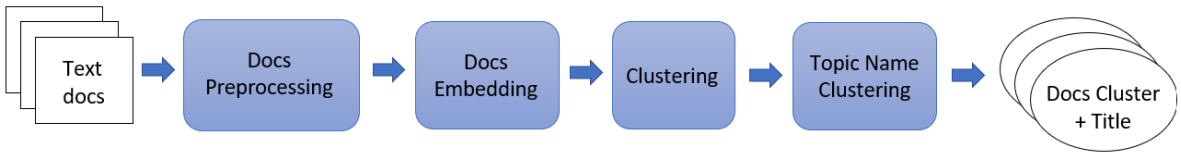

Figure 2: Pineline for Topic Clustering model.

2.2.1 Input: Text Documents

We built our own model to solve Topic Clustering problem. The model has the steps described in Figure 2. The input of our model is a set of text, total of 7532 documents, labeled with 20 target names.

2.2.2 Docs Preprocessing

Preprocessing text data is important in NLP because it helps standardize the text and prepare it for processing and analysis. The preprocessing steps that we use in our model are specifically (tokenizing, lemmatizing, removing short tokens, de-nonymizing, removing stopwords) to serve the following purposes:

1. Tokenizing: Splits text into individual words or phrases, making it easier to analyze and process.

2. Lemmatizing: Reduces words to their base form, which helps reduce the dimensionality of the data and makes it easier to process.

3. Removing short tokens: Helps remove words that are likely not meaningful or add little value to the analysis.

4. De-nonymizing: Replaces named entities with their semantic meaning, which can improve the results of NLP models.

5. Removing stopwords: Removes common words that are unlikely to provide meaningful information, such as "the," "and," "is," etc.

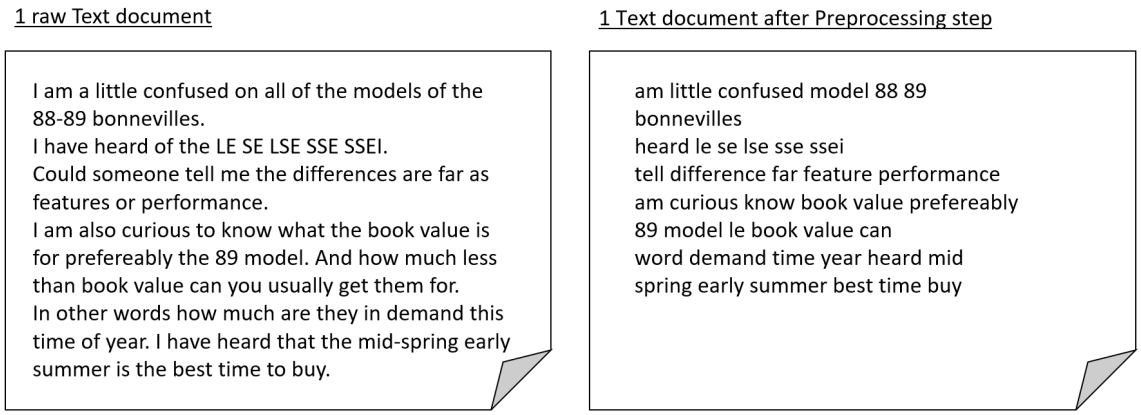

Figure 3: Example of a text before and after Preprocessing

2.2.3 Docs Embedding

We use Sentence Transformer as a pre-trained model for embedding documents.

SentenceTransformer("all-MiniLM-L6-v2") is a pre-trained language model that can be used to obtain sentence embeddings, which are numerical representations of the meaning of a sentence. The "all-MiniLM-L6-v2" version of SentenceTransformer has been trained to provide embeddings of the highest quality, using layer 6 of the MiniLM architecture.

Documents are passed through the SentenceTransformer("all-MiniLM-L6-v2") model, the output received will be a set of vectors with 384 dimensions. The values of each dimension range from -1 to 1 .

2.2.4 Clustering

After Docs Embedding step, we receive input as a set of vectors. In this step, we need to assign these vectors to clusters. By using Cluster Model.

Cluster model is a machine learning technique used to group similar data points (or "cluster" them) together based on their features or attributes. The goal of clustering is to partition the data into groups so that items in the same group are more similar to each other than those in other groups.

In our model, we use 3 different types of clustering models:

1. HDBSCAN: Hierarchical Density-Based Spatial Clustering of Applications with Noise. An algorithm to cluster data into groups based on density and similarity. [2]

2. OPTICS: Ordering Points To Identify the Clustering Structure. Algorithm to extract clusters in a database, characterized by high-density regions separated by low-density regions. [1]

3. KMeans: Partitioning method for clustering data into groups based on minimizing the sum of squared distances between data points and cluster centroids. [4]

2.2.5 Topic Name Clustering

In this step, we use BERTopic combine with Cluster model to cluster and assign title to each cluster. Here we have 3 combined methods including:

BERTopic is a deep learning algorithm that can be used in title clustering to group similar titles together. It works by encoding the meaning of titles into fixed-length vectors, allowing for effective comparison and clustering using distance metrics like cosine similarity. [3]

1. BERTopic + Cluster model HDBSCAN

2. BERTopic + Cluster model OPTICS

3. BERTopic + Cluster model KMeans

2.2.6 Output : Docs Cluster + Cluster title

The result of Topic Name Clustering step is the output of our model. Output includes docs clusters and cluster title corresponds to docs cluster.

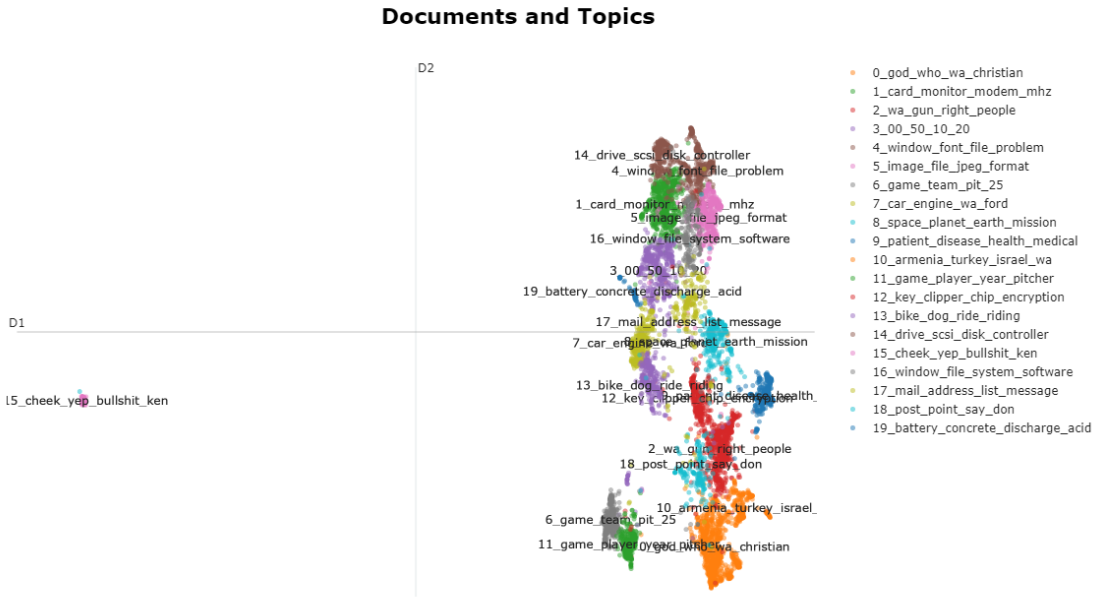

Figure 4: Image visualization of cluster and cluster title with our test data

2.3 Evaluation

Clustering is an unsupervised learning algorithms. But evaluation for unsupervised learning algorithms is a bit difficult and requires human judgement. To find suitable Topic Clustering model for the Hot Event Detection problem, We need some metric to evaluate the performance of each combined model.

Because our data is labeled data so we can use metrics that give us an idea of how good our clustering model is. However, these assessments are objective only and do not fully measure the level of model effectiveness we have built. In this project, we use 4 metrics to evaluate the quality of Clustering model include Mutual Information, V-measure, Completeness Score.

We used BERTopic combined with Cluster model including:

1. BERTopic + Cluster model HDBSCAN

Metric	Score
Mutual Information	1.2320
V-measure	0.3700
Completeness Score	0.3356

Table 2: Performance on test set with BERTopic and HDBSCAN.

2. BERTopic + Cluster model OPTICS

Metric	Score
Mutual Information	1.2359
V-measure	0.3513
Completeness Score	0.3055

Table 3: Performance on test set with BERTopic and OPTICS.

3. BERTopic + Cluster model KMeans

Metric	Score
Mutual Information	1.3810
V-measure	0.4663
Completeness Score	0.4710

Table 4: Performance on test set with BERTopic and KMeans.

Through the experimental results in three models, the model with the highest performance in 4 metrics is BERTopic with Cluster model KMeans. The second highest accuracy model is BERTopic with Cluster model HDBSCAN. The model with the lowest accuracy is BERTopic with Cluster model OPTICS. Therefore, we will choose model BERTopic with Cluster model KMeans and BERTopic with Cluster model HDBSCAN as the pipeline for the Hot Event Detection model.

3 Hot Event Detection

3.1 Hot Event Detection for News

Hot Event Detection refers to the process of identifying significant or breaking news events in real time by analyzing a large volume of data sources, such as news articles, social media

posts, and other relevant sources. This technology enables media organizations, businesses, and governments to quickly detect, understand and respond to emerging events, trends and public opinion in a timely and effective manner.

After we have built the Topic clustering model, we will use the most promising model to apply to the Hot Event Detection problem for News articles. To solve this problem, we need to combine 2 modules. The first module is the one that crawls news articles from the internet. The second module is the Topic Clustering model we built.

3.2 Model

3.2.1 News Crawler Module

To get data on the content of articles used for event analysis, we built a module to automate the data collection of internet journal articles. Specifically, we collect news sources from 2 famous news agencies, The New York Times news and BBC news.

News Crawler module requires the input of this module to enter a date from the user. The date entered by the user will be the date that the module uses to automatically collect data on news agencies. User can enter one or more dates according to need. A list of News will be collected from popular newspapers such as BBC news and NYT news based on date. These News will be processed and extracted content and titles. Output of Crawler News module will be a list of news content and titles.

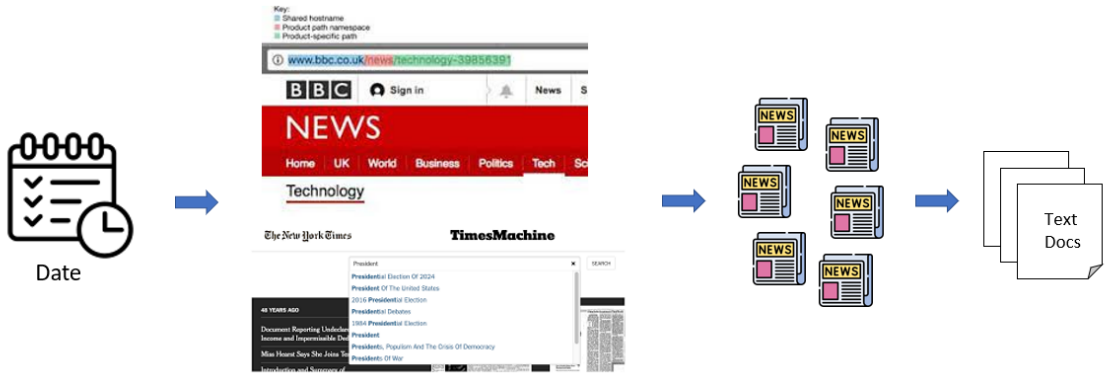

Figure 5: Pineline for News Crawler module.

3.2.2 Topic Clustering Module

In section 2 Topic Clustering, we experimented and evaluated BERTopic models combined with Cluster Models such as HDBSCAN, OPTICS, Kmeans. We will use the best matching model to make the pipeline for the Hot Event Detection problem.

We will combine the Topic Clustering module include BERTopic + Kmeans Cluster model with News Crawler module to cluster and detect hot events from the content of the news.

3.3 Result

To demonstrate the usefulness of our model, we run experiments with real data on the internet. We collected News from 09/01/2023 to 01/15/2023 (7 days total) with statistics as shown in the table below.

We run our model. The first one is module that combine Topic Clustering module include BERTopic with Kmeans Cluster model with News Crawler module to cluster and detect

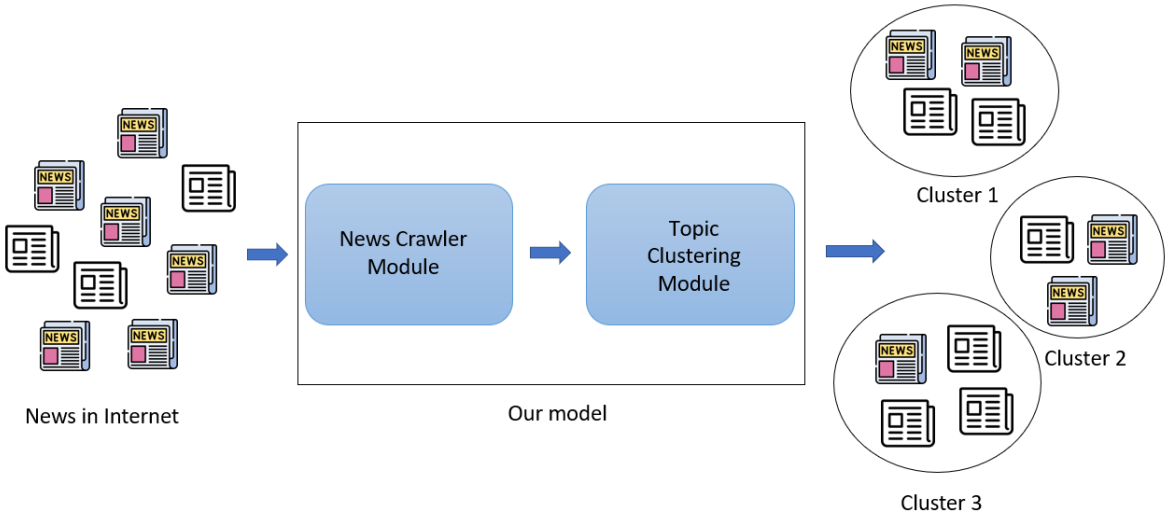

Figure 6: Pineline for our model.

Source	Quantity
The New York Times	829
BBC news	430
Total	1259

Table 5: Statistics of news data crawl from internet.

hot events from real data from internet. The second one is BERTopic with Cluster model HDBSCAN with News Crawler module.

The resulting image show that we have cluster with title in the right. Each cluster of data points has its own color and is plotted in 2D space. Because this is real data without labels, it is not possible to evaluate the performance and accuracy of the results.

Because the model solves Hot Event Detection problem, the output must be able to identify events from real data. At the time of data collection, the world had real hot events such as the war between Russia and Ukraine, the gas and energy shortage crisis in Europe, disaster in California, USA,...

- Model BERTopic with K-means Cluster generates 20 clusters. Model also identified these events in cluster 10 (ukraine russia soledar force), cluster 12 (climate gas energy europe), cluster 8 (california storm rain flood).
- Model BERTopic with Cluster model HDBSCAN generates 57 clusters include these events in cluster 0 (ukraine russia soledar force), cluster 3 (california storm rain water). Both models accurately predict hot events in the given time period.

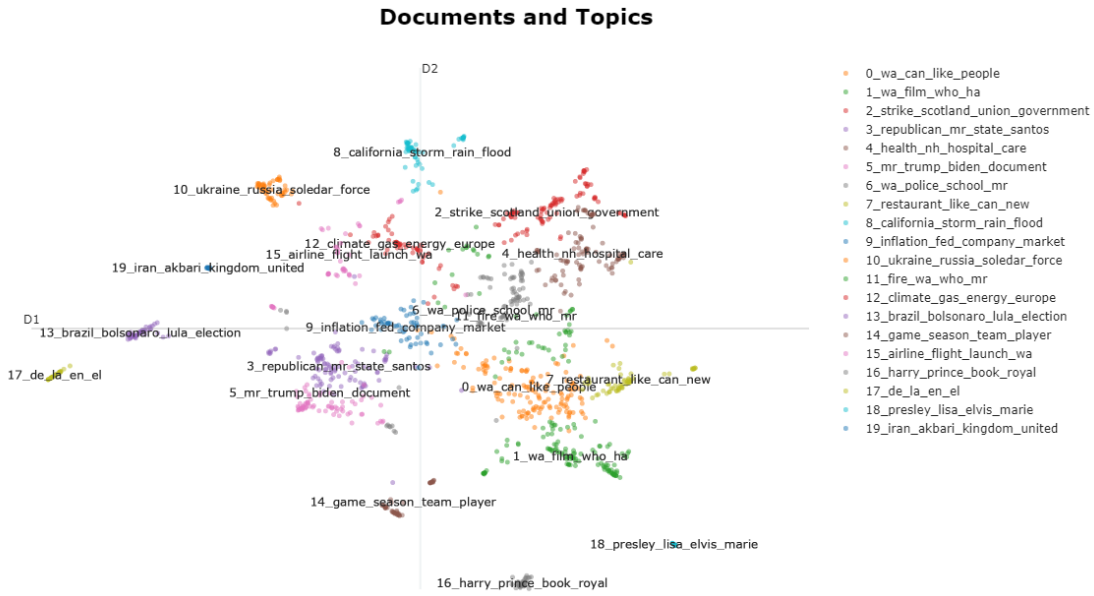

Figure 7: Image visualization of cluster and cluster title with real data using BERTopic with Kmeans Cluster

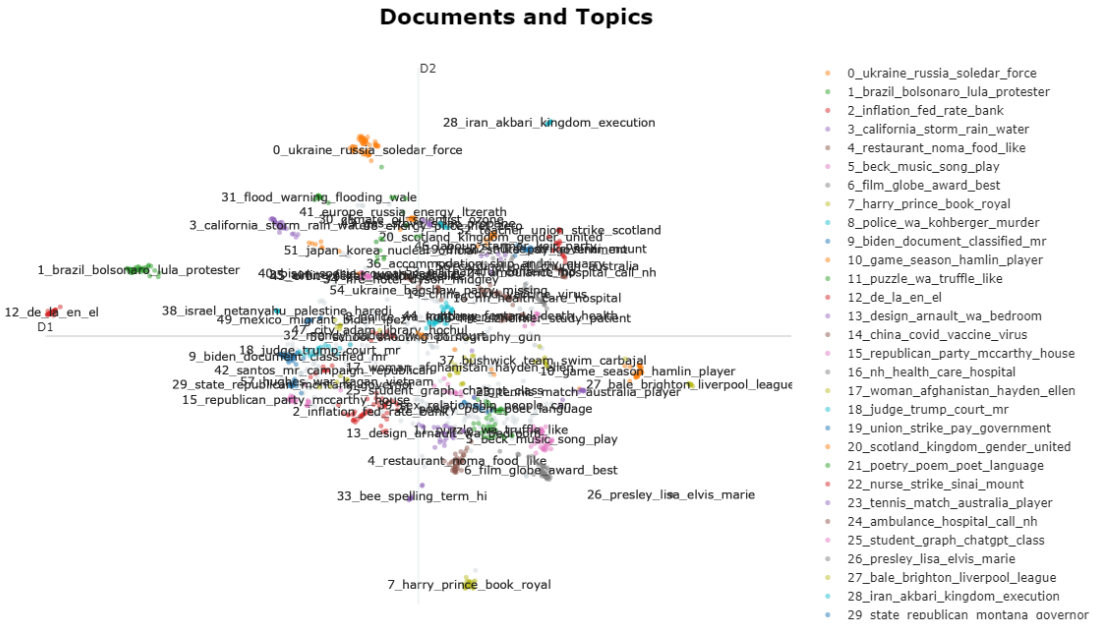

Figure 8: Image visualization of cluster and cluster title with real data using BERTopic with Cluster model HDBSCAN

However, with model BERTopic with Kmeans Cluster, due to the use of Kmeans algorithm, the number of clusters needs to be defined in advance. This makes it difficult to use in practice because depending on the amount of data and the nature of the data, there will be a different number of topics. Therefore, the cluster number parameter assigned first will sometimes be less or more than the number of topics of the actual data. Another problem is that, because Kmeans algorithm is absolutely clustered, that is, each data point will be assigned to a certain cluster, without outlier. Leads to misassigned data points.

With model BERTopic with HDBSCAN Cluster, with an unlimited number of clusters, the model can generate the suitable number of clusters for the actual data, which the Kmeans algorithm cannot. The data points are also classified into the correct cluster, not misassigned cluster as when using Kmeans. However, because the number of clusters depends on the actual data, sometimes the number of clusters generated is very large, larger than the number of topics in the actual data, resulting in many clusters with overlapping content.

In general, each model has its own pros and cons. The performance of these 2 models is quite good and can be applied in practice to predict real-life events. Depending on the amount and nature of the data and purpose, user can choose the appropriate model.

References

[1] Mihael Ankerst et al. "OPTICS: Ordering Points to Identify the Clustering Structure". In: vol. 28. June 1999, pp. 49–60. DOI: 10.1145/304182.304187.

[2] George D. Greenwade. *The Comprehensive Tex Archive Network (CTAN)*. 1993.

[3] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[4] Youguo Li and Haiyan Wu. "A Clustering Method Based on K-Means Algorithm". In: *Physics Procedia* 25 (Dec. 2012), pp. 1104–1109. DOI: 10.1016/j.phpro.2012.03.206.