School of Information Technology and Engineering at the ADA University

School of Engineering and Applied Science at the George Washington University

LARGE SCALE CLASSIFICATION AND CLUSTERIZATION OF COVID-19 RELATED PAPERS

A Thesis
Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Rustam Talibzade

April, 2023

THESIS ACCEPTANCE

This Thesis by: Rustam Talibzade
Entitled: *Large Scale Classification and Clusterization Of Covid-19 Related Papers*

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

(Adviser)	15/05/2022 (Date)
(Program Director)	15/05/2022 (Date)
(Dean)	15/05/2022 (Date)

ACADEMIC INTEGRITY STATEMENT

"I affirm that this is my own work, I attributed where I used the work of others, I did not facilitate academic dishonesty for myself or others, and I used only authorized resources for my Thesis, per the ADA University Academic Integrity requirements. If I failed to comply with this statement, I understand consequences will follow my actions. Consequences may range from failing the course to expulsion from the program/university and may include a transcript notation."

Rustam Talibzade		24/04/2023
(Full Name)	(Signature)	(Date: DD.MM.YY)

ABSTRACT

The year 2020 is mostly associated with the outbreak of COVID-19 caused by SARS-CoV-2 coronavirus due to immeasurable effects on our lives. The humanity faced unexpected challenges that were not faced in recent history. Plenty of research was done to find out the ways to combat COVID-19 disease and save as many lives as possible. This led to the emergence of huge number of articles and research papers in COVID-19 related literature, which were hard to keep up with. Several datasets like LitCovid and CORD-19 were created where collections of COVID-19 related literature is stored. To gain benefits and insights from such datasets, there is a need for data analytics and machine learning techniques to analyze these datasets.

This Master Thesis research explores a comprehensive analysis of text classification and clustering methodologies including Support Vector Machines (SVM), Naive Bayes, Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Non-negative Matrix Factorization (NMF), BERT, BioBERT, and SciBERT, applied to a large dataset of COVID-19 research articles sourced from the LitCovid database. The primary goal of this research is to devise and assess techniques for organizing, analyzing, and understanding the swiftly expanding collection of scientific literature pertaining to COVID-19.

The research is structured into multiple phases. Initially, a thorough literature review is conducted to establish a robust understanding of the cutting-edge developments in NLP, text classification, clustering and topic modelling. This review encompasses traditional machine learning techniques including supervised and unsupervised clustering algorithms. Their applications on different datasets including COVID-19 related datasets like CORD-19 and LitCovid are also discussed.

Next, description of LitCovid dataset is provided. Afterwards, machine learning techniques mentioned above are applied using different word vectorization techniques including Bag-Of-Words, TF-IDF and Word2Vec to identify how certain algorithms behave with these vectorization methods.

In the results and analysis section, the author offers a comprehensive comparison of all classification, topic modelling and clusterization approaches used for COVID-19 research articles. Finally, in the summary and future work section, the author consolidates key findings and considers potential work for future research.

TABLE OF CONTENTS

TABLE OF CONTENTS ..... **V**

LIST OF FIGURES ..... **VII**

LIST OF TABLES ..... **VIII**

1 INTRODUCTION **1**

1.1 Definition of the Problem 2
1.2 Objective of the Study 2
1.3 Significance of the Problem 2
1.4 Research Limitations 3

2 LITERATURE REVIEW **4**

2.1 Naïve Bayes Algorithm 7
2.2 Support Vector Machines 7
2.3 Latent Dirichlet Allocation 8
2.4 Non-Negative Matrix Factorization 9
2.5 Latent Semantic Analysis 10
2.6 BERT 10
2.7 SciBERT 11
2.8 BioBERT 11
2.9 K-Means 12

3 APPROACH AND METHODOLOGY **15**

3.1 LitCovid database 15
3.2 Dataset Description 15
3.2 Evaluation metrics 16
3.2.1 Accuracy *16*
3.2.2 Recall *16*
3.2.3 Precision *16*
3.2.4 F1-Score *16*
3.2.5 Macro-averaged evaluation metrics *17*
3.2.6 Micro-averaged evaluation metrics *17*
3.2.7 Weighted evaluation metrics *17*
3.2.8 Confusion Matrix *17*
3.2.9 Cross-Validation *18*
3.3 Text Preprocessing 18
3.4 Word Vectorization 18
3.4.1 TF-IDF *18*
3.4.2 Bag-Of-Words (Count Vectorizer) *19*
3.4.3 Word2Vec *19*
3.5 Topic Modelling 19
3.5.1 Latent Dirichlet Allocation *20*
3.5.2 Non-Negative Matrix Factorization *20*
3.5.3 Latent Semantic Analysis *20*
3.6 Classification 20
3.6.1 Multinomial Naïve Bayes *21*
3.6.2 Support Vector Machines *21*
3.6.3 BERT *21*
3.6.4 SciBERT and BioBERT *22*
3.7 Clusterization 22
3.7.1 K-Means *22*
3.8 Tools used 22

4. RESULTS AND DISCUSSION **23**

4.1 Classification results 23

4.1.1 Multinomial Naïve Bayes *23*
4.1.2 Support Vector Machines *24*
4.1.3 BERT, SciBERT and BioBERT *25*
4.2 Topic Modelling and Clustering 27
4.2.1 Latent Dirichlet Allocation *27*
4.2.2 Non-negative Matrix Factorization *28*
4.2.3 Latent Semantic Analysis *29*
4.2.4 K-Means *30*
4.3 Comparison 31
5 CONCLUSION AND FUTURE WORKS **32**
REFERENCES **32**
APPENDICES **34**
A.1 GitHub repository 34

LIST OF FIGURES

Figure 1. Bayes stratification by varying lengths of the tweet [19].....4
Figure 2. The topics extracted from the implementation of topic modeling algorithm on Iranian publications on Covid-19 in LitCovid [8]. 5
Figure 3. A support vector machine decision function (solid line) demonstrating the margin separating it from neighboring training samples.[3] 8
Figure 4. The intuitions behind latent Dirichlet allocation [10]. 9
Figure 5. LDA results example [10]. 9
Figure 6. Overall pre-training and fine-tuning procedures for BERT [15] 11
Figure 7. Overview of the pre-training and fine-tuning of BioBERT [27].....12
Figure 8. The outcomes of a typical clustering algorithm and a visualization of the cluster centers. [33] (left).....13
Figure 9. K-Means clustering and boundaries [34] (right) 13
Figure 10. Random centroids' initialization (left), Assignment of data points to nearest centroids(right) [34].....13
Figure 11 New centroids (left), assignment of data points to new centroids (right) [34].....14
Figure 12. An overview of the LitCovid daily workflow [5] 15
Figure 13. Confusion matrix for MNB 23
Figure 14. Confusion Matrices for SVM 24
Figure 15. BERT confusion matrix 26
Figure 16. SciBERT confusion matrix 26
Figure 17. BioBERT confusion matrix.....27
Figure 18. Cross-tabulation matrices of LDA assigned topics with real topics. 28
Figure 19. Cross-tabulation matrices of LSA assigned topics with real topics. 29
Figure 20. Cross-tabulation matrices of NMF with real topics. 30
Figure 21. Cross-tabulation matrices of K-Means with real topics. 30

LIST OF TABLES

Table 1: Topics mapping to integer 20
Table 2: Multinomial Naive Bayes classification report 23
Table 3: Support Vector Machines classification report 24
Table 4: BERT, SciBERT and BioBERT classification reports 25
Table 5: LDA Topics and ten most frequent words in each topic 27
Table 6: MNF Topics and ten most frequent words in each topic 28
Table 7: LSA Topics and ten most frequent words in each topic 29

LIST OF ABBREVIATIONS

Abbreviation	Explanation
WHO	World Health Organization
CORD-19	COVID-19 Open Research Dataset Challenge
CSET	Center for Security and Emerging Technology
AI2	Allen Institute for AI
OSTP	Office of Science and Technology Policy
NLM	The National Library of Medicine
CZI	The Chan Zuckerburg Initiative
CeDAR	Center of Data Analytics Research
NLP	Natural Language Processing
LDA	Latent Dirichlet Allocation
TF-IDF	Term Frequency-Inverse Document Frequency
MNB	Multinomial Naïve Bayes
LSA	Latent Semantic Analysis
SVM	Support Vector Machine
NMF	Non-Negative Matrix Factorization
SVD	Singular Value Decomposition
BERT	Bidirectional Encoder Representations from Transformers
MLM	Masked Language Model
NSP	Next Sentence Prediction
NIH	National Institutes of Health
PM	PubMed
PMC	PubMed Central
NER	Named Entity Recognition
RE	Relation Extraction
QA	Question Answering

NCBI	National Center for Biotechnology Information
NLM	National Library of Medicine
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
BoW	Bag-of-Words
TF	Term Frequency
IDF	Inverse Document Frequency
CBOW	Continuous Bag of Words

1 INTRODUCTION

The outbreak of Coronavirus disease 2019 (COVID-19) caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in the end of 2019 changed the lifestyle of the humanity and has had a significant impact on global health, economies, and many sectors of daily human lives. The disease that originates from a Chinese city of Wuhan spread with a high speed to other Chinese provinces, and later outside of the country, eventually reaching almost all parts of the world. In several months it became clear that COVID-19 may become a serious threat for humanity and on the 11th of March 2020 the outbreak was classified by the World Health Organization (WHO) as a pandemic.

The humanity faced a tough unexpected challenge, where it was hard to make any decisions because of lack of understanding of what the consequences may be. To understand and fight the disease, scientists from various fields did a lot of research and have published countless articles about different aspects of the virus and the disease, including its causes, transmission, prevention, treatment, and impact on different areas of our lives, where they were discussing their research, observations, and results. Consequently, it is not a surprise that the number of publications related to COVID-19 began growing quickly and despite the fact that the rate of infection cases decreases, the number of research papers and articles published about this topic still keeps growing and it becomes more difficult for scientists, medical professionals and analysts to keep up with the latest findings, and the most recent news.

One way to manage this large collection of articles is by using powerful methods to organize, sort, and summarize the information. To combine all these research papers in one place several datasets were created that store COVID-19 related literature. The most notable ones are COVID-19 Open Research Dataset Challenge (CORD-19) released by Allen Institute for AI (AI2), in collaboration with The White House Office of Science and Technology Policy (OSTP), the National Library of Medicine (NLM), the Chan Zuckerburg Initiative (CZI), Microsoft Research, Kaggle and Georgetown University's Center for Security and Emerging Technology (CSET) [1] and LitCovid database by National Center for Biotechnology Information of U.S National Library of Medicine.

Such paper collections have been used by medical professionals to conduct systematic reviews, that discuss COVID-19-related issues such as infection and mortality rates in various demographics, signs and symptoms of the illness, management strategies, interactions, and connections with other diseases [1]. These papers hold lots of useful information that can be used in many fields of our lives. While it seems that the pandemic is coming to an end, there are occasions when COVID-19 has long time effects. Also, during the pandemic lockdowns happened in many countries, where people were forced to stay at homes to avoid the further spread of the COVID-19. There was a need to adapt to these new realities and change the way they used to do different things. This is how online technologies began prospering even more during this period as people were staying indoors. More and more people started working from home, shopping online became more common thing to do. Furthermore, it is essential to analyze the mistakes and actions done with positive outcomes to consider them for the future and be ready to any similar challenges we may face.

Many studies were done on these different topics and can be investigated to benefit from their results and professionals in these fields will be able to benefit from these studies. However, medical and other researchers face difficulties in keeping up with such number of studies, locating appropriate papers in historical coronavirus literature, extracting useful information and grabbing maximum knowledge from the literature.

Machine learning, especially text classification and clustering techniques, can be useful for processing and analyzing the ever-growing body of COVID-19 scientific literature. The purpose of this paper is to look at applications of both supervised and unsupervised machine learning methods on COVID-19 literature. For this Thesis research the dataset mainly from the LitCovid database was used, which includes COVID-19 related publications divided to certain topics. The author applied and tested different classification and clusterization algorithms to identify which algorithms work best on the literature. Comparing the results of classification and clustering will help us better understand the structure of the research field and possibly find new connections between publications and faster ways to combine knowledge.

First step was review of other studies on this topic and find out how machine learning has been used before to manage and understand complex scientific literature. The algorithms were searched and analyzed of how they have been applied in various fields, especially in scientific and medical fields

including in COVID-19 literature. After reviewing the literature, the author describes the dataset, preprocessing methods, feature extraction techniques, and machine learning algorithms used for both classification and clustering. Next, the outcomes of the experiments were presented, including the visualization of the results, compared them, discuss interesting patterns and relationships found in the analysis.

1.1 Definition of the Problem

Categorizing and understanding the vast amount of COVID-19 research papers is critical for researchers, healthcare professionals, and general public to be in line with up to date COVID-19 research. Accurate classification and clustering of these documents it easier to synthesize knowledge, spot emerging trends, and identify potential research areas. However, manual classification of thousands of papers is a laborious, time-consuming, expensive, and error-prone task. This is particularly true when dealing with a rapidly evolving research landscape, such as the COVID-19 pandemic. As a result, there is a pressing need for automated techniques that can efficiently and accurately classify and cluster research papers, making them more accessible and interpretable.

1.2 Objective of the Study

The first goal of this study is to find the most optimal model for classification of COVID-19 research papers using machine learning techniques. Classification, which is a supervised approach will allow us to assign research papers to one of the predefined categories. This involves training different classifier models with different vectorizing techniques, evaluating their performance, and finding the most accurate one. By leveraging automated classification methods, the aim is to improve the efficiency and accuracy of identifying and categorizing research papers.

The second goal of this study to try different clusterization algorithms and evaluate performances of different approaches. Clustering involves grouping research papers based on their similarities without using predefined labels, offering an unsupervised approach to data organization.

The final goal is to compare and match the classification results with clustering outcomes. By comparing classification and clustering results, the goal is to uncover patterns and insights that may not be immediately apparent through classification alone.

1.3 Significance of the Problem

Before deciding to make a certain important action, it is essential to do research on the given situation. This is especially important when it is about human health and medicine. Classifying and clustering of research papers hold significant potential in aiding researchers in making better-informed decisions saving more of valuable time. COVID-19 being a serious issue is an area, that has a huge need for research and will be researched for many upcoming years. Large datasets like CORD-19 and LitCovid exists with hundred thousand of papers, which seems like a great place to explore. However, it is hard for a human being to read all that papers, also, a lot of them may not be helpful for a researcher on their given situation, which means much time will be spend on articles, that are not suitable for their research. Without a proper analysis, this huge chunk of data is not valuable as it can be.

Efficient categorization and organization of research findings can streamline the scientific process by facilitating rapid knowledge synthesis, reducing redundancy, and directing resources towards unexplored or under-investigated areas. By providing an automated means of supervised and unsupervised approaches on research papers, we can improve the speed and accuracy of information retrieval, making it easier for experts to access relevant studies and build upon existing knowledge. Furthermore, insights derived from classification and clustering can help in identifying emerging trends and areas requiring further investigation, ultimately accelerating scientific progress.

The main motivation for this research emerged after examining the CORD-19 dataset. As mentioned before, the dataset has a lot of research papers and articles that can be valuable for certain people. However, it is difficult to find relevant papers based on the topic you search for. This is when the decision was made to develop a model for classifying and clustering of COVID-19 related papers.

1.4 Research Limitations

LitCovid database is a huge dataset that contains full texts for the most of its articles. Due to its size technical limitations, it was hard to train machine learning models on the author's laptop, so computers in Center of Data Analytics Research (CeDAR) in ADA University were used. However, due to the university not being on certain days and hours, the available time of using these computers was limited.

2 LITERATURE REVIEW

Literature Review is the first step of any research to prepare for its implementation part. This section provides information about existing research that was done on text classification, clustering, and natural language processing (NLP) in the field of scientific, biomedical and COVID-19 related literature analysis.

Methods and algorithms used for this Thesis are part of Machine Learning. To understand it is important to understand what Machine Learning is. According to Goodfellow et al. [2], Machine Learning can be seen as the practical application of statistical methods that rely heavily on computer algorithms to estimate complex functions. Thus, the focus is more on accurately predicting outcomes and less on proving confidence intervals around those predictions. In Machine Learning we can point out two types of learning methods: supervised and unsupervised. Supervised learning aims to make predictions about an outcome measure by analyzing various input measures. On the other hand, unsupervised learning does not involve any outcome measure, but instead focuses on identifying patterns and associations among a given set of input measures [3].

In order to classify COVID-19-related tweets into positive, negative, and neutral categories, Ramya et al. [19] adopted logistic regression and Naïve Bayes classifiers. Using a package called NLTK, he has classified Tweets according to sentiments such as anxiety and sadness. These sentiment scores are separated into train and test data, stratification techniques based on machine learning are applied, and results are discussed. In his research paper, the system architecture includes the creation of a dataset in which tweets are transmitted using the Python NLTK library and stored in a database. The tweets are then pre-processed, which includes data cleaning that concentrates primarily on removing abusive words ('abvs') and common words such as 'the', 'a', 'an', etc. Following preprocessing, the data are given to a trained classifier, which classifies tweets as positive, strongly positive, neutral, negative, moderately negative, or strongly negative. As training data, approximately 10,000 tweets were collected, and as test data, approximately 1,000 tweets were collected. The Twitter API is used to retrieve test data, which is then fed to the trained classifier for sentiment analysis. Because of its support for Naïve Bayes classifiers based on Gaussian distribution, NLTK is a very strong and widely used library in Python. The collated tweets are converted to csv format in order to be processed using Python. These tweets are used to categorize the sentiment into positive, negative, and neutral categories. Following numerous iterations, the Naive Bayes classifier returned the number of occurrences of each word that indicated positivity, negative, or neutrality towards COVID-19. His result shows that Naïve Bayes impart 92% and 60% accuracy for different number of characters. [19]

Tweets (ncharacters < 70)				**Tweets (ncharacters < 150)**			
	Positive	**Neutral**	**Negative**		**Positive**	**Neutral**	**Negative**
Positive	40	1	5	Positive	6	1	30
Neutral	5	2	1	Neutral	5	2	1
Negative	1	1	43	Negative	1	1	43
Accuracy: 0.9249				Accuracy: 0.6056			

Figure 1. Bayes stratification by varying lengths of the tweet [19]

Damanik, F J, and Setyohadi, D B, published a similar study in the 2020. For their research the Multinomial Naive Bayes method was employed. This method is used to determine the highest public opinion average regarding Covid-19. The obtained results are the number of scores, the average value from precision and recall, with the highest result indicating a decent or excellent method for analyzing the problem. 5000 data were analyzed using the Multinomial Naive Bayes method and also Support Vector Machine, which yielded a positive response or opinion from the Indonesian population regarding Covid-19, with 2004 positive responses, 999 negative responses, and 1997 neutral responses. The figure

1 from the paper shows that based on the calculated average score, the Support Vector Machine approach was shown to be the most effective model in this investigation by a significant margin of 93%. The Multinomial Naive Bayes approach only differs by 2%.

To analyze a group of texts, Sperandeo et al. [10] utilized topic modeling that aims to identify the main topics or themes present within the texts. In their opinion developing a topic model determines the level of similarity among concepts. This refers to the relationship between different words that goes beyond their literal usage and meaning, allowing for a more nuanced understanding of the underlying topics present in the texts. Their research topic was "What Does Personality Mean in the Context of Mental Health?". They applied Latent Dirichlet Allocation (LDA) for their research. Their first step was determining the number of topics using a cross-validation method. Different numbers of topics were extracted from the dataset, and their performance was evaluated by comparing the perplexity, which is a statistical parameter that compared the training values during the cross-validation, and the one with lowest perplexity was taken because it is the indication of the model being effective in making predictions. After applying LDA, they got 30 topics, which they analyzed and discussed.

Dastani and Danesh [8] applied text mining and topic modelling techniques to Iranian COVID-19 publications that are presented in LitCovid. Firstly, the relevant documents were selected, and the words used in these texts were extracted. Next, the texts were unified by manually analyzing the keywords used in the articles, and by identifying and removing synonymous words and meaningless words or stop-words. The most important words were then identified using the Term Frequency-Inverse

LitCovid topic	Subtopics based topic modeling	Keywords
General	Topic1: General[1]	—
Mechanism	Topic1: Characteristics	Case, virus, people, smoking, potential, expression, disease, spread, find, level
	Topic2: Clinical features	Cell, infection, clinical, virus, case, acute, respiratory, severe, disease, lymphocyte
	Topic3: Genomic sequence	Sequence, respiratory, virus, phylogenetic, isolate, infection, mutation, strain, cause, acute
Transmission	Topic1: Environment	Rate, case, temperature, fatality, cluster, increase, spread, attack, correlation, size
	Topic2: Different areas	Transmission, disease, infection, rate, region, case, outbreak, report, measure, sample
	Topic3: Modes	Case, transmission, travel, cluster, age, early, global, report, individual, infection
Diagnosis	Topic1: Infection	Infection, case, disease, risk, chest, severe, clinical, symptom, evaluate, report
	Topic2: Risk factors	Mortality, clinical, outcome, risk, disease, diabetes, hospital, age, death, factor
	Topic3: Symptoms	Case, symptom, age, clinical, mortality, disease, positive, infection, test, confirm
Treatment	Topic1: Clinical features of mortality	Clinical, mortality, case, risk, hospital, factor, value, confirm, report, age
	Topic2: Clinical features of the disease	Disease, clinical, case, mortality, outcome, severe, infection, risk, age, rate
	Topic3: Drug	Control, trial, treatment, clinical, cell, drug, vitamin, receive, level, disease
	Topic4: Outcome	Trial, clinical, treatment, outcome, control, intervention, participant, arm, randomization, registration
Prevention	Topic1: Behaviors	Pandemic, health, disease, care, risk, control, outbreak, dental, infection, intervention
	Topic2: Management	Pandemic, hospital, health, stroke, knowledge, disease, management, base, care, infection
	Topic3: Policy	Pandemic, public, health, people, infection, policy, risk, disease, model, case
	Topic4: Control	Case, pandemic, disease, risk, control, spread, outbreak, infection, epidemic, people
	Topic5: Other diseases	Pandemic, outbreak, health, infection, report, care, trauma, increase, case, review
	Topic6: Pandemic status	Case, model, rate, death, spread, pandemic, measure, confirm, mortality, estimate
Case report	Topic1: Children	Case, clinical, infection, acute, test, child, report, syndrome, respiratory, neonate
	Topic2: New symptoms	Case, report, liver, manifestation, infection, symptom, clinical, cutaneous, disease, respiratory
	Topic3: Pregnant	Report, infection, case, old, pregnant woman, chest, treatment, refer, pandemic, hospitalize
	Topic4: Death	Report, case, disease, respiratory, acute, novel, rate, death, fetal, severe
Forecasting	Topic1: Estimate	Case, estimate, model, epidemic, death, predict, confirm, trend, outbreak, base
	Topic2: Modeling	Model, case, spread, rate, confirm, death, predict, pandemic, prediction, trend
	Topic3: Epidemic	Model, case, estimate, trend, base, policy, disease, infection, epidemic, prediction
	Topic4: Spread	Case, base, pandemic, spread, region, weather, mean, humidity, temperature, estimate

Document Frequency (TF-IDF) method, which measures the importance of a word in a document or a set of documents. Prior to extracting the most important words based on TF-IDF, the words used in the text were stemmed using the Porter stemmer algorithm. For the topic modelling itself, Latent Dirichlet

Figure 2. The topics extracted from the implementation of topic modeling algorithm on Iranian publications on Covid-19 in LitCovid [8].

Algorithm (LDA) was used. As LDA does not determine the number of topics, Dastani and Danesh used the logarithmic UMass Coherence criterion to identify the appropriate number of topics. This criterion proposed several values for the number of topics; thus guidance was sought from medical specialists to determine the optimal number of topics for each category. As a result of their research, they found out that most of the global and Iranian publications were in "Prevention" category of LitCovid database; "COVID", "patient", and "Iran" were the most frequent words in Iranian publications of COVID-19 on LitCovid; words "patient", "pandemic", "outbreak" were the most important words based on TF-IDF weights. Lastly, in Figure 2 they shared their results of the topic modelling on Iranian publications for each of the category on LitCovid (Long Covid category did not exist in LitCovid during their research, yet).

Khateeb [9] made several experiments on CORD-19 dataset as part of his Master Thesis. The objective of the thesis was to make a classification of the publications in CORD-19 dataset and forecast a publications' citation rates based on its abstract content. He used Multinomial Naïve Bayes (MNB) and Support Vector Classifier (SVC) models for the experiment. First of all, he searched for TF-IDF features in the papers and extracted them. Using TF-IDF the most prominent features were extracted and used, which allowed him to decrease training time of the model. As a result of this experiment, he found out that by taking alpha = 1 for MNB and linear kernel model for SVC the highest amount of accuracy can be achieved. Next, BERT and SciBERT tokenizers were used for documents' abstracts. After all the experiments, the accuracy, precision, recall, f1-score, training, and test times of all the models were compared. MNB was the fastest one, however SciBERT managed to achieve the highest accuracy, while being very slow to train and test.

Colavizza et al.'s topic modelling analysis was also done by concatenating titles and abstracts of CORD-19 papers into one string, like the previous study. Words with a high likelihood for the same topic have a tendency to co-occur often in the same papers, which is how a topic is determined considering probability distribution across a vocabulary. They first converted titles and abstracts to bag-of-words form using ScispaCy's en_core_sci_md model. Next, Latent Dirichlet Allocation (LDA) was applied. After this study it was determined that before SARS and MERS outbreaks the topics were mostly about molecular biology and immunology. After SARS and MERS outbreaks the topic of epidemics became popular, and since 2020 publications are mostly on COVID-19 pandemic, its effects and management [21].

Boyack et al. [14] compared different clustering algorithms to find out which similarity approach would produce the most consistent and accurate clusters for a set of over two million literature documents. The publications were taken from MEDLINE database. Their first step was preprocessing the texts in documents. They concatenated the title and abstract of each document, and then removed all punctuation marks except apostrophes from the text. The removed punctuations were replaced with a single space to ensure the text remained coherent and the structure of the text is not destroyed. The text was then lowercased and tokenized into tokens based on whitespace. The tokens that were empty or contained whitespace were discarded. Tokens with standard contractions like "don't," were separated into a root and contraction form, such as "do not." These contractions were then removed from the text since they appeared in their stopword lists or were possessive forms of other words. Afterwards, they started applying different similarity approaches. The first approach was finding cosine similarities of TF-IDF coefficients. More on how to calculate TF-IDF is described in Chapter 3.2. The other approach was Latent Semantic Analysis (LSA), which is a technique in NLP that examines the connections between a group of documents and the terms they contain, producing a set of concepts that are relevant to both the documents and the terms. This algorithm assumes that word that have similar meanings will appear in the document in similar situations within the text. Third approach was Poisson-based language model for ranking (BM25) which is rarely used for clustering and is more applicable in Information Retrieval systems, though Boyack et al. believe that this model is still suitable for large dataset like the one they are using. Their next approach was Self-organizing map which is a type of artificial neural network that transforms high-dimensional data into a low-dimensional geometric model. This method creates a grid of neurons, where each neuron has a vector that corresponds to a position in the term space. Unlike the input vectors, the neuron vectors have continuous weights for each term, rather than discrete counts. Initially, all the neuron weights are randomly assigned. During training, individual document vectors are presented to the neuron grid, and the neuron vector that is most like the input vector (using cosine similarity) is identified. The best-matching neuron and its neighboring neurons are

then pulled closer to the input vector. The degree of adjustment is determined by the grid distance between the best-matching neuron and its neighbors within a specific neighborhood diameter. This process is repeated until the neuron grid forms a geometric model that accurately represents the original high-dimensional data. Lastly, topic modeling was used which identifies and learns a group of thematic topics represented as lists of related words that describe a collection of texts. The model then assigns a small number of these topics to each document in the collection, enabling the researcher to identify the underlying themes present in the texts. Eventually, BM25 model showed being the most accurate model, followed by Topic Modeling [14].

2.1 Naïve Bayes Algorithm

The Bayes Theorem is the foundation of Naïve Bayes classification, which is a probabilistic classifier. The probability of a class is calculated for each sample. All characteristics are presumed to be independent. The classifier selects the classification that is most similar to V_{nb} with the provided attribute $a_1, a_2, a_3, \ldots, a_n$. V_{nb} can be calculated using the following formula: [17]

$$V_{nb} = argmax_{v_{j \in V}} P(v_j) \prod P(a_i|v_j)$$

Using the m-estimate, calculation of $P(a_i|v_j)$ happens by this formula:

$$P(a_i|v_j) = \frac{n_c + mp}{n + m}$$

Explanation of symbols:
n = the number of training examples for which v = v_j
n_c = number of examples for which $v = v_j$ and $a = a_i$
p = a priori estimate for $P(a_i|v_j)$
m_p = an equivalent sample size
m = the equivalent sample size

Multinomial Naïve Bayes (MNB) is a Naïve Bayes variant developed to address text document classification problems. MNB employs a multinomial distribution as a classification feature, using the number of repetitions of a word or the weight of the word. A multinomial event model represents each document by the set of word occurrences in the text. That is, the word sequence is not preserved. It produces the well-known container of words representation for documents. Each document may easily be seen as a histogram, with each element indicating the number of repetitions of the resultant word in the document. The MNB equation is represented as follows: [18]

$$P\ (X|c)\ =\ log\ \frac{Nc}{N}\ +\ \sum_{i=1}^{n} \log \frac{ti + \propto}{\sum_{i=1}^{n} t + \propto}$$

P ($X|c$) = probability document X in class c
Nc = total documents in class c
N = total documents
ti = weight term t
$\sum_{i=1}^{n}$ = total weight term t in class c
$\propto$ = smoothing parameter

2.2 Support Vector Machines

Support Vector Machine (SVM) is a popular topic of study in supervised learning classifier research, particularly for text classification. Support Vector Machine is one of the earliest supervised learning classification techniques, and it is widely used in a variety of fields, including handwritten digit recognition, bio informatics, facial recognition, and classifying texts. It employs a point-based representation of text examples in a multidimensional space. Then, new texts are classified based on their similarities to existing text and the regions to which they are mapped. They are robust in high-dimensional spaces and also perform well with sparse data. In addition, SVM has achieved success in the domain of opinion mining. [28] The immense expansion of IT also enlarges the text document

collection. Consequently, the storage space and computational cost of model learning are massive. Instance selection is one way of bypassing these limitations in order to avoid this issue. The classical SVM is systematically extended to the classification of multiple instances with distinct dimensions. [29]

Despite their advantages, support vector machines contain one or more parameters that must be exhaustively investigated in order to develop an optimal classifier. SVMs function by locating a decision function that attempts to divide the training data into two categories. The decision function is chosen so as to maximize the distance between it and the adjacent training data on either side of the surface. If no decision function can separate the data linearly, a kernel transformation function is used to map the data into a distinct dimensional space (called a feature space) so that it can be separated linearly using traditional support vector machine decision function techniques. Multiple varieties of kernels have been devised in order to map data into dimensions of varying sizes. A slack error variable is utilized to generate a soft margin decision function for data separation if the kernel transformation function does not entirely separate our data. This function ensures that our data are not mixed together. Figure 3 depicts an example of an SVM decision function and the margin. [30]

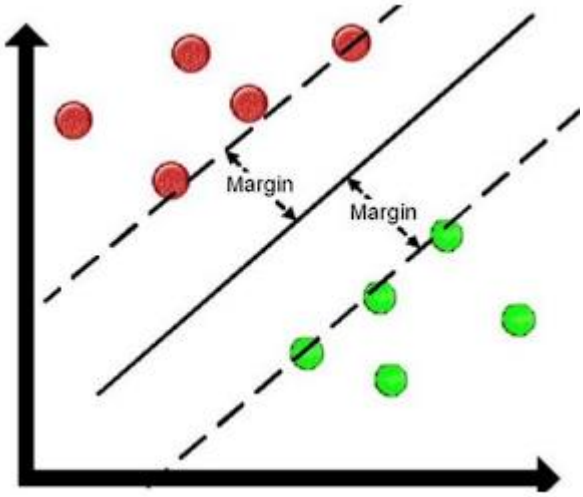


Figure 3. A support vector machine decision function (solid line) demonstrating the margin separating it from neighboring training samples.[3]

2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a nifty machine learning technique that helps us make sense of massive amounts of unstructured text data. It uncovers hidden topics within large document collections, even when the connections between the documents might not be obvious at first. LDA in its current form was proposed by Blei et al. [12] who described this model as a three-tiered hierarchical Bayesian model that represents each item in a collection as a finite combination of underlying topics. Each topic is then modeled as an infinite mixture of underlying topic probabilities. In the context of text modeling, these topic probabilities offer an explicit representation of a document.

In Figure 4 from another paper by Blei [10], he shares a sneak peek on LDA. He presumes that there are a certain number of "topics," which are word distributions, applicable to the entire collection (far left). Each document is assumed to be generated through the following process. First, select a distribution over the topics (the histogram on the right); then, for every word, assign a topic (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments on this figure are shown just for illustration and are not from the real data. In Figure 5, Blei shared how does the real results of LDA algorithm look like, where on the left you can find the topic proportions from the article of Figure 4 and on the right the 15 most frequent words from the most frequent topics can be found.

In simple terms, LDA assumes there are a certain number of concealed "topics" lurking within the documents. It presumes that each document is a blend of these topics. LDA believes that every topic is made up of words that frequently appear together. So, what the model does is essentially play detective,

trying to figure out which words belong to which topics and what proportion of each topic is found in every document. By doing this, LDA brings clarity to vast amounts of text data, revealing the hidden structure and relationships between documents.

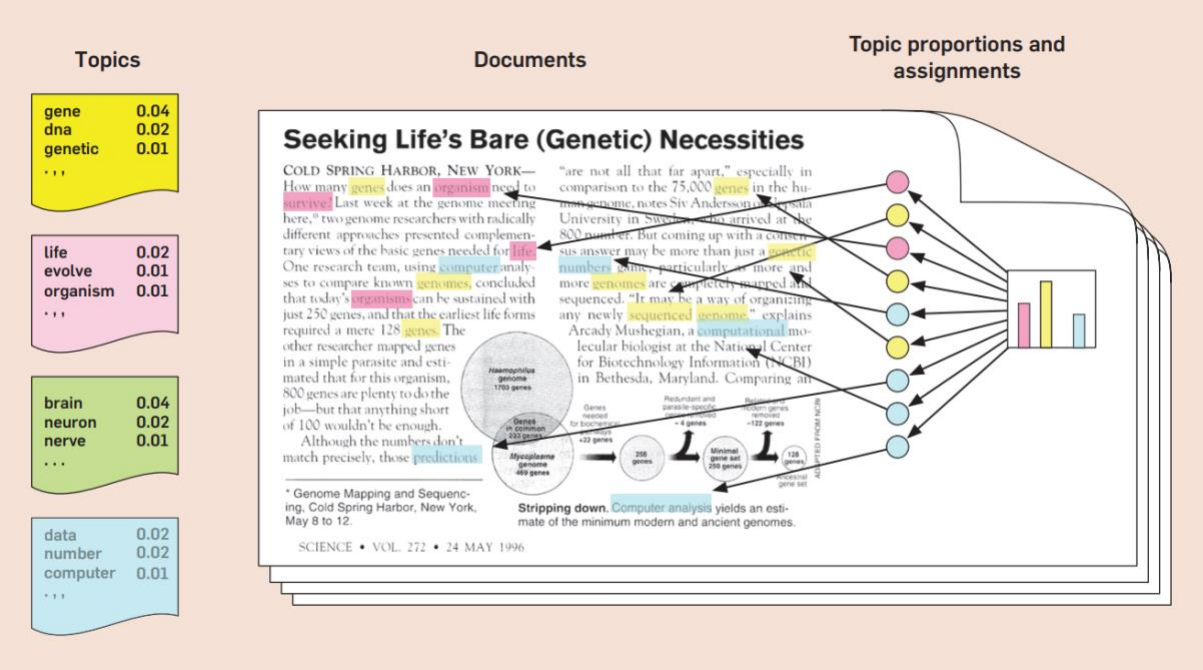


Figure 4. The intuitions behind latent Dirichlet allocation [10].

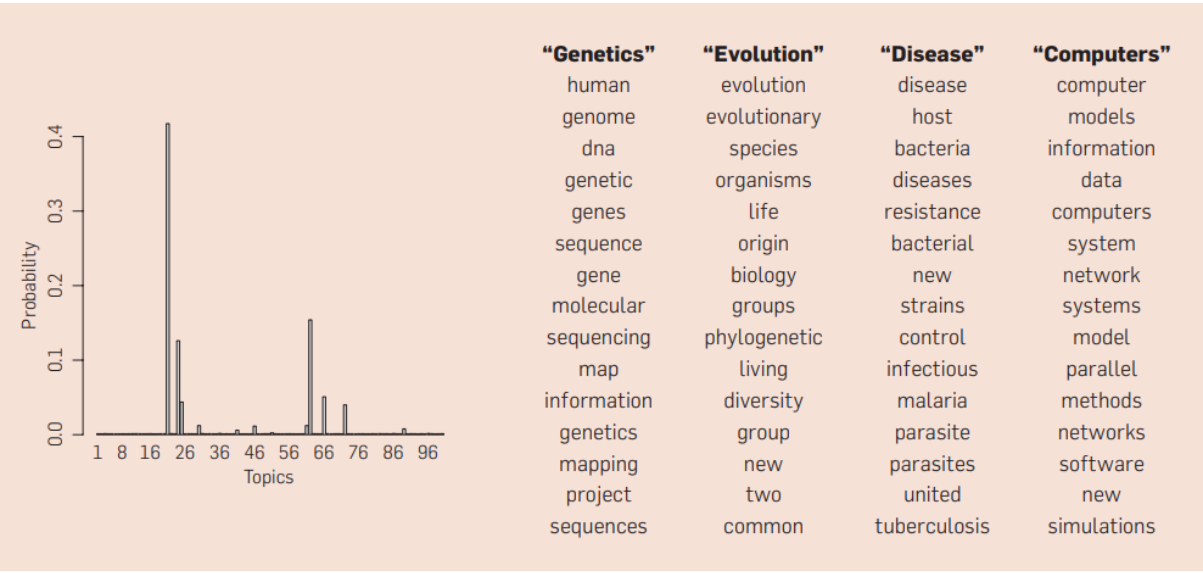


Figure 5. LDA results example [10].

2.4 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is a powerful statistical technique, NMF factorizes a non-negative matrix into two lower-dimensional matrices with non-negative entries, which approximate the original matrix when multiplied together. Non-Negative Matrix Factorization (NMF) is a statistical technique that helps reduce the size of large datasets. It works by analyzing the input data and giving less importance to less relevant information. Specifically, NMF factorizes an input matrix V into two smaller matrices, W and H, where W has a dimension of m x k and H has a dimension of n x k. In our case, V represents a term-document matrix, where each row of H represents a word embedding and

each column of W represents the relevance of each word to each sentence. All the entries in W, H, V should be positive. NMF is widely applied in topic modeling, where it is used to discover latent patterns and topics in large collections of documents. [31]

2.5 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a topic modeling technique that aims to identify the context in which a term appears in a corpus and mathematically model it using singular value decomposition (SVD). It involves developing a document-term matrix and approximating eigenvectors for each word using SVD, which are then compared to determine their similarity. The resulting word eigenvectors are factor rotated to create a nonoverlapping fit for the model, and topics are derived from the clustering of the word eigenvectors throughout the corpus. LSA requires less computing power than other topic modeling methods, but it cannot estimate a structure directly from raw data, and its level of analysis is further removed from the data compared to other methods. Therefore, researchers must carefully consider the benefits and tradeoffs of each method [32].

2.6 BERT

In 2018, Devlin et al. [15] introduced new language representation model – Bidirectional Encoder Representations from Transformers (BERT). The BERT model differs from other language representation models in that it is specifically designed to create bidirectional representations from unlabeled text by conditioning on both left and right context at all layers during pretraining. This unique design allows BERT to be fine-tuned with only one additional output layer to create highly effective models for various tasks, including language inference and question answering, without requiring significant adjustments to the model architecture for each specific task. One of the main techniques used by BERT is masked language models that are used to generate deep bidirectional representations during pretraining. By pre-training these representations, BERT can greatly reduce the need for complex, task-specific model architectures that were previously heavily engineered. According to Devlin et al. BERT's ability to be fine-tuned with just one additional output layer sets it apart as the first representation model to achieve state-of-the-art performance on a broad range of tasks, both at the sentence-level and token-level, surpassing the performance of many task-specific architectures.

There are two steps in BERT: pre-training and fine-tuning. In the pre-training phase of BERT, the model is trained on a variety of tasks that involve predicting masked words in a sentence or predicting the relationship between two sentences. This allows BERT to learn general language representations that can be fine-tuned for specific downstream tasks. The fine-tuning phase involves initializing the BERT model with the pre-trained parameters and then fine-tuning all the parameters with labeled data from downstream tasks. Even though each downstream task has its own fine-tuned model, all the models are initialized with the same pre-trained parameters. One of the distinctive features of BERT is its unified architecture across different tasks. There is minimal difference between the pre-trained architecture and the final downstream architecture, making it easy to fine-tune BERT for new tasks. BERT's model architecture is based on the original implementation of the Transformer encoder described in Vaswani et al. [16] and has multiple layers of bidirectional self-attention. The number of layers, hidden size, and number of self-attention heads can be adjusted to create different model sizes. BERT's bidirectional self-attention is a critical feature that sets it apart from other models like GPT, which use constrained self-attention where each token can only attend to the context to its left. This allows BERT to better capture the relationship between words in a sentence and generate more accurate representations for downstream tasks. The pre-training phase of BERT involves two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In the MLM task, a percentage of input tokens are randomly masked, and the model is trained to predict these masked tokens. The final hidden vectors of the masked tokens are fed into an output SoftMax over the vocabulary, similar to a standard LM. By masking tokens randomly, the model is forced to generate deep bidirectional representations and can learn the context of the masked token from both left and right contexts. To avoid creating a mismatch between pre-training and fine-tuning, the training data generator chooses 15% of the token positions at random for prediction. When a certain token is chosen, it is replaced with the masked token 80% of the time, a random token 10% of the time, and the unchanged token 10% of the time. This allows the model to learn how to handle out-of-vocabulary words and better generalize

to unseen data. The NSP task involves predicting if sentence B follows sentence A or not. For each pre-training example, the sentences A and B are chosen, and B is either the actual next sentence that follows A or a random sentence from the corpus. This task helps the model understand the relationship between two sentences which is beneficial for downstream tasks like Question Answering and Natural Language Inference. The pre-training corpus includes BooksCorpus and English Wikipedia. The corpus is document-level, not shuffled sentence level. For Fine-Tuning, in applications involving text pairs, BERT uses the self-attention mechanism to unify the encoding of text pairs and bidirectional cross attention. Fine-tuning involves plugging in the task-specific inputs and outputs and fine-tuning all parameters end-to-end. Figure 6 demonstrates the architecture for BERT model. The architecture used for pre-training and fine-tuning of BERT is the same, except for the output layers. Thus, the same pre-trained model parameters are used to initialize models for different downstream tasks during fine-tuning. All parameters are fine-tuned during the fine-tuning stage, allowing the model to be customized for specific tasks.

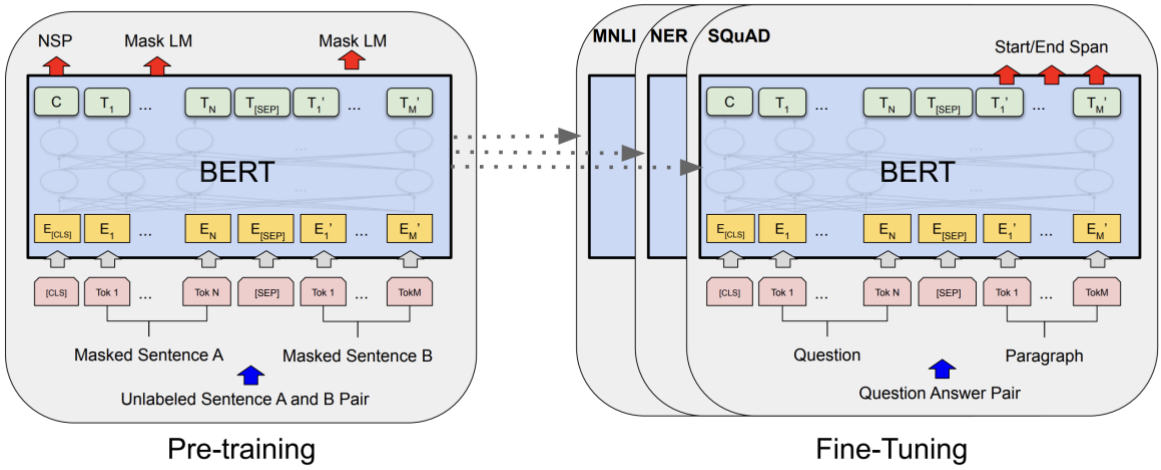


Figure 6. Overall pre-training and fine-tuning procedures for BERT [15]

2.7 SciBERT

SciBERT is a language model developed by the Allen Institute for Artificial Intelligence (AI2), SciBERT is a variation of BERT model, specifically tailored to the domain of scientific literature. The modifications made to the original BERT model have resulted in a highly specialized and efficient tool for handling scientific text.

Pre-training data of BERT mainly consists of general-domain text from sources like Wikipedia and the BooksCorpus. As a result, its domain-specific knowledge is limited, including the areas like scientific literature, where concepts prevail. Recognizing this limitation, researchers at AI2 crafted SciBERT to address the challenges inherent in scientific text analysis. They trained SciBERT on a massive corpus of scientific publications (more than 1.14 million) from the Semantic Scholar archive, covering the fields of computer science and biomedical domain. This training enabled the model to understand and generate text that is more representative of scientific literature, subsequently leading to improved performance in domain-specific NLP tasks.

Like the BERT model, SciBERT is also based on the transformer architecture, which allows it to effectively capture long-range dependencies and model context across large spans of text. The model is bidirectional, meaning it processes text both from left to right and from right to left, enabling a better understanding of the contextual relationships between words. Beltagy et al. state that their model outperforms BioBERT model for some tasks [26].

2.8 BioBERT

BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) is a language representation model developed by Lee et al. [27] from the Korea University and the

National Institutes of Health (NIH) that is designed to improve the performance of natural language processing (NLP) tasks on biomedical text. BioBERT is based on the BERT architecture.

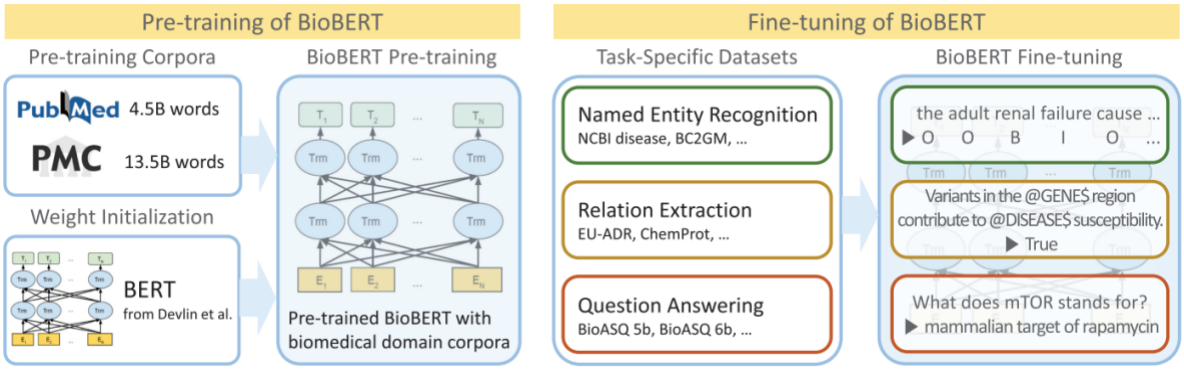
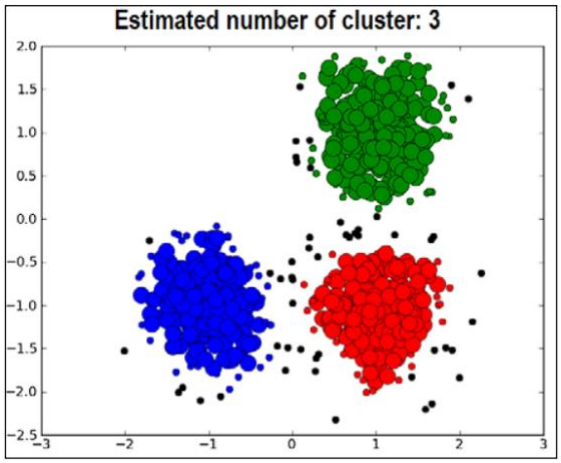


Figure 7. Overview of the pre-training and fine-tuning of BioBERT [27]

There is great number of domain-specific words that appear usually in biomedical text. As BERT was trained on Wikipedia and BooksCorpus, NLP models that are designed to understand general-purpose language often perform poorly when used in biomedical text mining tasks. To address this, the Lee et al. pre-trained BioBERT using PubMed abstracts and full-text articles from PubMed Central (PMC). In the Figure 7, the authors of the model showed their overview on how they pre-trained and fine-tuned their model. First, they started by initializing BioBERT with weights from BERT, which had previously been pre-trained on general domain corpora like English Wikipedia and BooksCorpus. Lee et al. then pre-trained BioBERT on biomedical domain corpora like PubMed abstracts and PMC full-text articles. Their next step was evaluation of the effectiveness of their approach by fine-tuning and testing BioBERT on three popular biomedical text mining tasks – Named Entity Recognition (NER), Relation Extraction (RE), and Question Answering (QA). The researchers tested different pre-training strategies using various combinations and sizes of general domain corpora and biomedical corpora and analyzed the effect of each corpus on pre-training. They also conducted in-depth analyses of BERT and BioBERT to explain the necessity of their pre-training strategies. Afterwards, BioBERT was fine-tuned on the tree above-mentioned tasks. Eventually, they compare BioBERT to BERT where they showed that BioBERT proves itself more accurate in the biomedical literature.

2.9 K-Means

Clusters are groups of items that are related to one another but distinct from objects in other clusters. When given a set of objects, clustering algorithms organize these objects into categories based on their similarities. The most prevalent application of clustering is to investigate the data and identify all possible meaningful categories. K-Means clustering algorithm locates the centroid of groups of data points. For the first time centroids are chosen randomly. To achieve accurate and efficient clustering, the algorithm evaluates the distance between each point and the cluster's center. Figure 8 illustrates how the K-means algorithm attempts to aggregate related data elements within the predetermined k = 3 clusters. [33] To divide the dataset, a measure of proximity must be defined. The Euclidean distance is the most prevalent measurement for a numeric attribute. Figure 9 depicts the aggregation of the Iris data set based solely on the petal length and petal width variables. This Iris data set has numerical properties and a k value of 3. The result of k-means clustering provides a distinct partition space for Cluster 1 and a confined partition space for Clusters 2 and 3. Initiating k randomly chosen centroids is the initial stage of a k-means algorithm. The user must provide the desired number of clusters, denoted by k. In this scenario, we create three data centers from scratch. For clarity, in Figure 10 we use a circle to denote each initial centroid and the same shape for data points that have been allocated to that centroid.



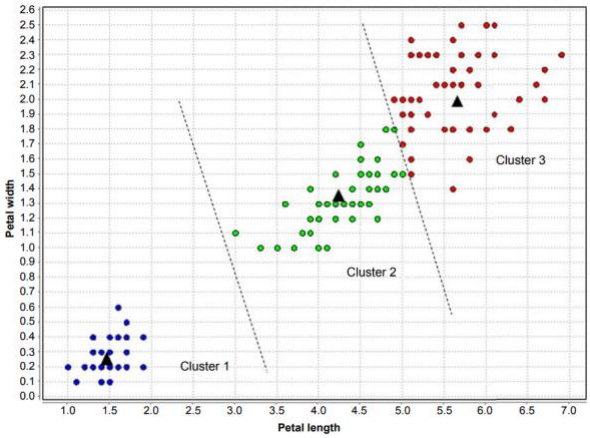


Figure 8. The outcomes of a typical clustering algorithm and a visualization of the cluster centers. [33] (left)

Figure 9. K-Means clustering and boundaries [34] (right)

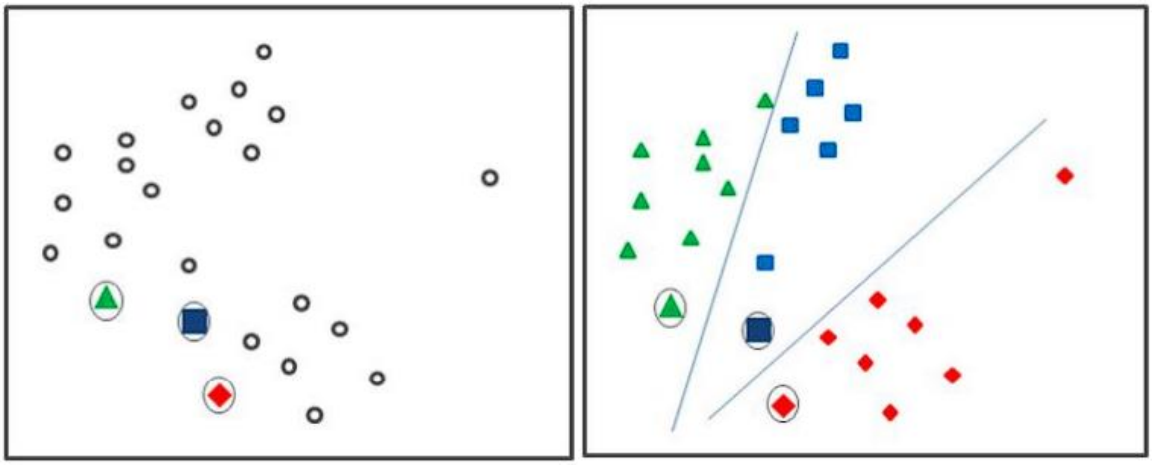


Figure 10. Random centroids' initialization (left), Assignment of data points to nearest centroids(right) [34]

Once centroids have been created, all of the data points are linked to the one that is geographically closest to it. Here, "nearest" is determined using some sort of distance measure. Although alternative metrics like the Manhattan measure and Jaccard coefficient can be chosen, Euclidean distance measurement is the most typical proximity measure. The Euclidean distance between two data points $X\ (x_1, x_2, \ldots x_n)$ and $C\ (c_1, c_2, \ldots c_n)$ with n attributes is shown as:

$$d = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \ldots + (x_n - c_n)^2}$$

A new centroid, serving as a mean example of each cluster, needs to be determined. This new centroid better represents the cluster than any other data point. Once the new centroids have been determined, the process of moving data points to the new centroid that is closest to them is repeated. Figure 11 shows how two previously unrelated data points were reclassified into the same cluster. After centroids are finally defined, when new data points are added to the dataset, they are assigned to the cluster with the smallest Eucledian distance [34].

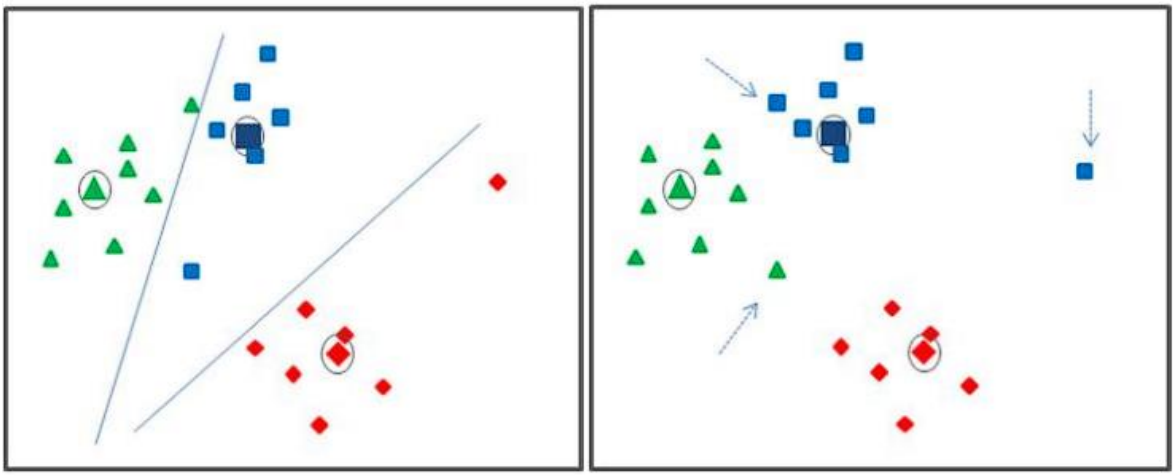


Figure 11 New centroids (left), assignment of data points to new centroids (right) [34]

3 APPROACH AND METHODOLOGY

3.1 LitCovid database

LitCovid is a publicly accessible database that allows users to freely download articles and their associated curated data for research discovery and machine processing purposes. The database was created by National Center for Biotechnology Information (NCBI), a division of National Library of Medicine (NLM) which is part of the United States National Institutes of Health (NIH) in response to the COVID-19 outbreak and is a great resource for researchers, healthcare professionals and public to access scientific publications on COVID-19.

Papers and articles in LitCovid are taken from PubMed, which is another database by the NLM. As of April 2023, LitCovid has more than 340 thousand publications from 8000 journals. With LitCovid you can also get links to full text of majority research papers and articles. Most of the publications are also categorized into 8 topics: Mechanism, Transmission, Diagnosis, Treatment, Prevention, Long Covid, Case Report, Epidemic Forecasting. LitCovid website which has a search engine, allows its users to search and filter publications based on the above-mentioned topics, related countries, journals, SARS-CoV-2 variants, vaccines, and drugs. The articles in LitCovid are specifically focused on COVID-19.

Curation of LitCovid was initially happening manually, however, as outbreak evolved and with the rapid growth of SARS-CoV-2 and COVID-19 literature, there was a need to support manual curation with automated approaches. In the Fig.12, you can find an overview of LitCovid's daily workflow. Firstly, articles from PubMed are extracted using this query: 'coronavirus'[All Fields] OR 'ncov'[All Fields] OR 'cov'[All Fields] OR '2019-nCoV'[All Fields] OR 'COVID-19'[All Fields] OR 'SARS-CoV-2'[All Fields]. Next, their relevance to COVID-19 is reviewed and only relevant publications are kept. To aid human reviewers, machine learning methods like Support Vector Machines (SVM) using bag-of-words features and word embedding based convolutional neural networks are used. After relevant publications are selected, they are annotated for topics, geolocations, journals, coronavirus variants, vaccines, and drugs. Topic assignment happens with the help of deep learning model with the embeddings created by BioBERT and later is examined manually. Eventually, the publications are indexed and sent to a server, where it can be later accessed from a web application. [4-7]

A subset of the dataset with abstracts of publications is available in CSV format, while the whole database is available in JSON/XML format. The drawback is that the dataset is packed in one huge JSON/XML file, which complicates text mining a bit. Especially it became difficult, when it was discovered that there is an issue with reading the JSON file, as there is a line of text in the file which is not a part of JSON format. However, that problematic line was found and fixed, thus the research continued with the updated JSON file.

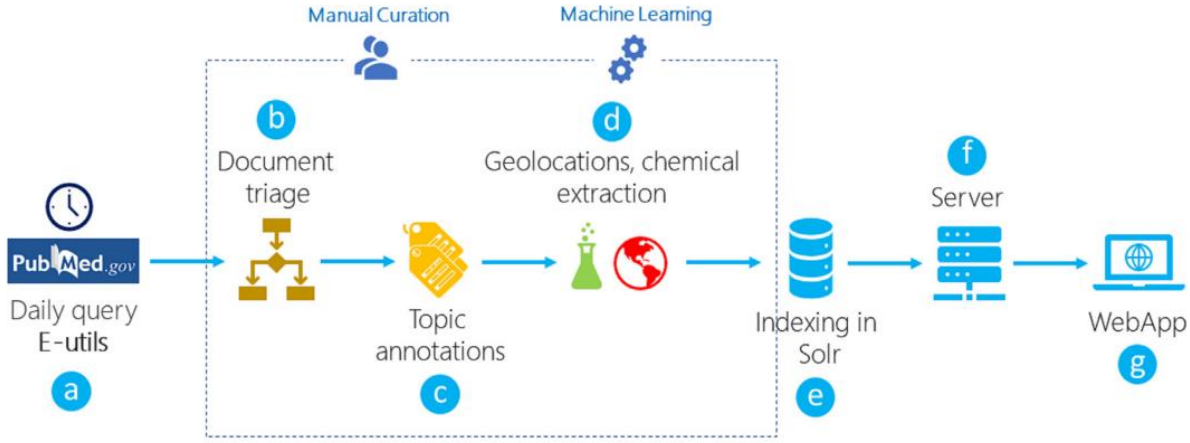


Figure 12. An overview of the LitCovid daily workflow [5]

3.2 Dataset Description

The JSON file includes most of the metadata about research papers and articles including title, authors, PubMed (PM) id, PubMed Central (PMC) id (if exists), partial or full texts for the publications, authors

of the publication. While majority of the needed data can be found in LitCovid's JSON file, the topics (labels) that publications are assigned are not in it, which are very crucial for my research. The labels for papers and articles were taken directly from LitCovid's web application.

Publications in LitCovid are usually in English, however sometimes you can find papers in other languages. For this research publications only in English were chosen. Some articles lacked both abstracts and full texts in the JSON file. For these papers, abstracts of the papers were taken from LitCovid web application. However, even on LitCovid website there were articles that did not have any text or abstract, so they were skipped for the research. In those publications that have full texts, the paper is divided into different sections. For example, many papers have chapters like Introduction and Results, it is possible to extract these specific chapters if needed.

It's worth noting that despite the extensive search the author could not find any codes that can read the JSON file of LitCovid database. Thus, the file reader was written from scratch which can be found in the author's GitHub repository together with the topics extractor.

For this research, around 22 thousand papers were used. First, not all papers are labeled in LitCovid, so they were not considered for this research as it would not be possible to evaluate the accuracy for these papers. There are some papers in LitCovid that are given several topics. This researched was focused on one-label classification, thus only publications with one topic were chosen for the research. Due to very low number of papers with "Transmission" topic, this topic was dropped, and research was conducted with the remaining seven topics.

3.2 Evaluation metrics

In this section various evaluation metrics are described including those that have been used to evaluate the performance of the Supervised Learning algorithms in my research. This includes recall, precision, accuracy, f1-score among others.

First, it is important to get acquainted about the concepts of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). TP and TN are correctly predicted positive and negative classes correspondingly, while FP and FN are wrongly predicted positive and negative classes.

3.2.1 Accuracy

Accuracy is the proportion of correct predictions made by a classifier in comparison to the total number of instances and is calculated by formula shown in (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

For this research the formula needs to be modified as there are 7 different labels, thus the formula (2) is used.

$$Accuracy = \frac{number\ of\ correctly\ predicted\ labels}{number\ of\ all\ examples} \qquad (2)$$

3.2.2 Recall

Recall measures the proportion of true positive instances that were correctly identified by the classifier.

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

3.2.3 Precision

Precision is the proportion of true positive instances among all instances classified as positive.

$$Precision = \frac{TP}{TP+FP} \qquad (4)$$

3.2.4 F1-Score

The F1-score is the harmonic mean of precision and recall, aiming to find a balance between these two metrics. It is especially useful when the dataset is imbalanced or when both false positives and false negatives carry significant costs.

$$F1\ score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (5)$$

As our model has more than two classes, the formulas (3) – (5) need to be modified:

3.2.5 Macro-averaged evaluation metrics

Macro-averaging calculates the evaluation metric separately for each class as if it were a binary classification problem, and then computes the arithmetic mean of the metrics across all classes. Macro-averaging treats all classes as equally important and is not influenced by class imbalance. To get macro-averaged recall, precision, f1-score formulas (6) – (8) are used correspondingly.

$$Macro\ Recall = \frac{\sum_{i=1}^{n} Recall_i}{n} \qquad (6)$$

$$Macro\ Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \qquad (7)$$

$$Macro\ F1\ score = \frac{\sum_{i=1}^{n} F1\ score_i}{n} \qquad (8)$$

In our case n = 7, as we have 7 classes.

3.2.6 Micro-averaged evaluation metrics

In micro-averaging we sum TP, FP and FN for all classes and then compute the evaluation metric as in a binary classification problem. To get micro-averaged recall, precision, f1-score formulas (9) – (11) are used correspondingly.

$$Micro\ Recall = \frac{\sum TP}{\sum TP + \sum FN} \qquad (9)$$

$$Micro\ Precision = \frac{\sum TP}{\sum TP + \sum FP} \qquad (10)$$

$$Micro\ F1\ score = \frac{2 \cdot Micro\ Recall \cdot Micro\ Precision}{Micro\ Recall + Micro\ Precision} \qquad (11)$$

3.2.7 Weighted evaluation metrics

Weighted metrics calculate the evaluation metric separately for each class as in a binary classification problem and then compute the weighted average of the metric values, where the weights are proportional to the number of instances in each class. They consider both class distribution and classifier performance on individual classes. Weighted metrics are helpful when assessing the model performance with the accent on large classes while still considering the performance on the smaller ones. To get weighted recall, precision, f1-score formulas (12) – (14) are used correspondingly.

$$Weighted\ Recall = \frac{\sum_{i=1}^{n} (Recall_i \cdot N_i)}{\sum_{i=1}^{n} N_i} \qquad (12)$$

$$Weighted\ Precision = \frac{\sum_{i=1}^{n} (Precision_i \cdot N_i)}{\sum_{i=1}^{n} N_i} \qquad (13)$$

$$Weighted\ F1\ score = \frac{\sum_{i=1}^{n} (F1\ score_i \cdot N_i)}{\sum_{i=1}^{n} N_i} \qquad (14)$$

Where n is number of classes and N number of samples of the class $\boldsymbol{i}$.

3.2.8 Confusion Matrix

A confusion matrix is a table that allows us to better understand the performance of the algorithm by displaying the number of true positive, true negative, false positive, and false negative predictions. For this research confusion matrices will be in form of 7x7 tables, because of the number of labels. Each element (i, j) in the matrix represents the number of instances of class i was predicted as class j. So, when $i = j$ it means the prediction was correct.

3.2.9 Cross-Validation

Cross-validation is a technique used in machine learning to evaluate the performance and generalize the ability of a model. It helps assess how well a model will perform on unseen data and minimize overfitting. Overfitting occurs when a model learns the noise in the training data instead of the underlying patterns, which can lead to poor performance on new data.

In cross-validation, the dataset is divided into k smaller subsets or "folds." The model is then trained on k-1 of these folds and tested on the remaining fold. This process is repeated k times, with each fold being used as the test set exactly once. The model's performance is then averaged across the k iterations to provide a more robust estimate of its performance. Cross-validation is beneficial because it helps to prevent overfitting, provides better estimates of model performance, and aids in model selection and hyperparameter tuning.

3.3 Text Preprocessing

Text preprocessing is the first step in NLP and text analytics tasks. The goal here is to convert raw text data into a well-structured and clean format to make it easier for machine learning algorithms to process and comprehend. This phase is of importance since the quality of the preprocessed data directly affects the outcomes of the following classification and clustering operations. Within the scope of my Master Thesis, preprocessing is a key step in ensuring that COVID-19 related papers are properly prepared for effective classification and clustering based on their content.

The goal of text preprocessing is to remove any noise, inconsistencies, or irrelevant information that may hinder the performance of the subsequent analysis. Common preprocessing techniques include tokenization, stopword removal, case normalization, stemming, and lemmatization, among others.

For preprocessing of texts, the 'en_core_sci_lg' model from ScispaCy library was used, a popular NLP library in Python. This model is a pipeline specifically designed for biomedical texts, thus, provides better results for domain-specific terminology, also has a huge vocabulary and vectors for 600k words. All words were lowercased for case normalization and were lemmatized. Then stopwords and punctuations were filtered. Furthermore, only words with a length of more than 3 were kept. This decision was made due to many papers, which had formulas with many standalone letters. They were making the data too noisy and corrupting the training of machine learning models that were used.

To sum up this section, text processing was a very crucial step in converting research papers and articles to clean format.

3.4 Word Vectorization

Word vectorization is a crucial process in natural language processing and machine learning that involves converting raw textual data into numerical representations. By converting words or phrases into vectors, algorithms can process the textual data more effectively, enabling tasks such as clustering, classification, and similarity detection. Word vectorization techniques capture the semantic and syntactic relationships between words, allowing the machine to identify patterns and make data-driven decisions. Various techniques were used for word vectorization during the Thesis research, including Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec. The choice of technique depends on several factors such as the problem, available data, and level of complexity. While simpler tasks may require BoW or TF-IDF, more complex tasks may benefit from the richer representations provided by Word2Vec. Ultimately, the effectiveness of machine learning algorithms in processing and learning from textual data depends on the quality of word vectorization. For this research, various Machine learning algorithms were tested with various word vectorization techniques.

3.4.1 TF-IDF

TF-IDF means term frequency – inverse document frequency. It is basically a measure that shows significance of the words in a document for the given dataset. TF-IDF can be found by multiplying the values of term frequency (TF) and inverse document frequency (IDF), which is shown in (15).

$$TF - IDF(term, document, dataset) = TF(term, document) \cdot IDF(term, dataset) \qquad (15)$$

Term frequency as it can be understood from the name shows how frequent is the given term within the given document and can be calculated with the formula shown in (16).

$$TF = \frac{number\ of\ times\ the\ given\ term\ occurs\ in\ document}{total\ number\ of\ terms\ in\ the\ document} \quad (16)$$

IDF is a measure that shows how many documents have the given term. It can be calculated using the formula shown in (17).

$$IDF = log\left(\frac{number\ of\ all\ documents}{number\ of\ documents\ containing\ the\ given\ term}\right) \quad (17)$$

TF-IDF Vectorizer was very useful in converting words into the format that is understandable by machine learning models. To sum up, what has been stated above TF-IDF is a word vectorization method that considers not only how often a word appears in a document, but also its rarity across the entire collection of documents. The concept is that words that are commonly found in a particular document but are infrequently used across other documents are likely to be significant and useful in understanding the document's content.

3.4.2 Bag-Of-Words (Count Vectorizer)

The Bag of Words (BoW) model or in another words Count Vectorizer is a simple word vectorization technique that represents text as a "bag" of words while disregarding the order or grammar of the words. The BoW model creates a fixed-size vocabulary from the input text corpus and represents each document as a feature vector of the same size as the vocabulary. Each element in the vector denotes the frequency or presence of a word from the vocabulary in the document.

Despite its simplicity, BoW can be quite useful in various machine learning and natural language processing tasks, including document classification. However, the model has its limitations, including its inability to capture context or word relationships, as the order of words is disregarded [22].

3.4.3 Word2Vec

Word2Vec is one of the most popular word vectorization techniques nowadays in NLP and was developed by Google in 2013. It is a powerful word vectorization technique that leverages neural networks to create dense word embeddings, capturing semantic and syntactic relationships between words. Unlike BoW and TF-IDF, which produce sparse, high-dimensional vectors, Word2Vec generates low-dimensional, continuous vector representations that can effectively capture word associations and contextual information. Word2Vec consists of two main algorithms: Continuous Bag of Words (CBOW) and Skip-Gram. CBOW predicts a target word based on the context words in each window, while Skip-Gram predicts the context words based on the target word. By learning to make these predictions, Word2Vec creates dense word embeddings that encapsulate semantic and syntactic relationships, which can then be used for various NLP tasks, including clustering, classification, and analogical reasoning. One drawback of Word2Vec is the need for large amounts of training data to create meaningful embeddings, although pre-trained models can help alleviate this requirement [23-24].

To create Word2Vec model, first, the text was tokenized. The vector size was set to 100, which means each word in the model is represented by a vector of 100 dimensions. Window size was chosen to be equal to 3, meaning the model considered a window of 3 words to the left and 3 words to the right of the target word in each sentence. 250 epochs were set, so that the Word2Vec model train 250 times over the tokenized dataset.

3.5 Topic Modelling

Topic modelling is a powerful unsupervised machine learning technique used to discover hidden patterns and structures in large collections of textual data, like research papers. Using topic modelling, it is possible to identify and group papers that share similar content, providing an organized and structured overview of the given dataset. For this Thesis, different topic modelling algorithms were applied on LitCovid publications. The goal of this task was to analyze whether Topic Modelling algorithms can spot different topics in our dataset. Initially, this task was applied to CORD-19 dataset

which has publications not only about COVID-19, but also other coronaviruses like SARS-CoV and MERS and to define the number of topics that would be used for topic modelling, coherence score was calculated using different number of topics to get the best number of topics. Eventually, when LitCovid database was chosen as the main dataset for this Thesis and the goal of the research became the comparison with different algorithms applied, it was decided to take same number of topics as used for classification – 7. The hypothesis before the start of the research was that it will be possible to differentiate those topics using Topic Modelling techniques.

3.5.1 Latent Dirichlet Allocation

The first algorithm tested was Latent Dirichlet Allocation (LDA). As vectorization techniques Count Vectorizer and TF-IDF Vectorizer were used. Due to Word2Vec vectors having negative values in them, this vectorization technique cannot be applied with LDA. Scikitlearn's LatentDirichletAllocation() method was used for implementation. Number of components (which means how many topics will be derived) was given the value of seven. LDA allocated each paper to certain topic and afterwards, 10 most frequent words for each topic were printed.

3.5.2 Non-Negative Matrix Factorization

The second topic modelling algorithm used was Non-Negative Matrix Factorization (NMF). Both BoW and TF-IDF Vectorizer were used as vectorization techniques, Word2Vec was omitted due to having negative values in the vectors. Scikit-learn's NMF() method was used, then seven topics with 10 most frequent words were printed. Later, each paper is assigned one of the topics.

3.5.3 Latent Semantic Analysis

Lastly, Latent Semantic Analysis (LSA) was evaluated. Like the previous BoW and TF-IDF vectorization techniques, and as TF-IDF giving better results it was kept. Scikit-learn's Pipeline() function was used where a Pipeline object with TF-IDF Vectorizer and Truncated SVD was created. After fitting the dataset, topics were defined and ten most frequent words from each topic was printed.

3.6 Classification

Classification, a fundamental task in machine learning and natural language processing, aims to assign predefined categories (labels) to a given input based on its features. In the context of research papers, classification models can be used to automatically categorize documents according to their research domains, methodologies, or other relevant attributes, providing a structured overview of the COVID-19 research landscape. This thesis examines the application of various classification models to COVID-19 research papers sourced from the LitCovid database. The goal is to identify and compare the performance of different classification methods in terms of accuracy and efficiency, with the ultimate goal of determining the most appropriate approach for organizing and analyzing the ever-growing body of COVID-19 literature. The study used various classification techniques, including traditional models such as Naïve Bayes algorithm, support vector machines (SVM), alongside more advanced deep learning methods like transformers, more specifically BERT and SciBERT.

Due to BERT accepting only integer values as inputs, the integer representation of topics was used for all classification algorithms to be able to compare all these models together. Table 1 shows the mapping of topics (labels) assigned to research papers and their mapped integer value.

Table 1: Topics mapping to integer

Topic	Integer value
Case Report	0
Epidemic Forecasting	1
Long Covid	2
Mechanism	3
Treatment	4

Topic	Integer value
Prevention	5
Diagnosis	6

3.6.1 Multinomial Naïve Bayes

The first classification method applied was Multinomial Naïve Bayes algorithm (MNB). To compare and find out which vectorization method works best with MNB Count Vectorizer and TF-IDF Vectorizers were used. As with LDA, due to negative values in Word2Vec vectors, it could not be used with MNB, either. First of all, the dataset is divided into train and test sets with the ratio of 80:20. For all the machine learning techniques that were used afterwards the same train and test sets were applied for comparison. Then, cross-validation was applied to evaluate the training process. 10% was allocated to cross-validation set. Next, the MNB model was trained on the training dataset and predictions were made on test dataset. For the model assessment classification_report() method from scikit-learn library was used which calculates and prints precision, recall, f1-score and accuracy of the model.

3.6.2 Support Vector Machines

Support Vector Machines was the second supervised learning algorithm that was used during this study. The linear kernel was used for this research. After fitting the training data to SVM, to get the best regularization parameter Grid Search was applied. Grid search is a common method in machine learning to enhance a model's performance by optimizing its hyperparameters. For Support Vector Machines (SVM), it is necessary to determine the best combination of hyperparameters, including the regularization parameter (C) and kernel function parameters. Grid search involves creating a list of potential values (for this experiment the list of potential values consisted of 0.1, 1 and 10) for each hyperparameter and then systematically explores every possible combination. The combination that results in the highest performance metric is considered the optimal set of hyperparameters for your model and then is chosen for training SVM on the dataset. Grid search already includes cross-validation inside of it, so there is no need of doing another cross-validation.

After the model is trained the predictions are made on the test data. Lastly, classification report with evaluation metrices is printed and confusion matrix of true and predicted labels are plotted. All the vectorization methods that were mentioned in Vectorization section were used on SVM including Word2Vec technique.

3.6.3 BERT

The next step in the thesis research was to test BERT. PyTorch and HuggingFace libraries were used to help with the model. However, due to BERT's weight, fine-tuning the model required a significant amount of time and computational power. To speed up this process, the help of a GPU was required, and the computers in CeDAR were used for this purpose. These computers have NVIDIA RTX A5000 GPUs with 24GB GPU memory installed in them.

To prevent wasting time waiting for BERT to finish fine-tuning, a small subset of the dataset was used for initial trial runs. This allowed for the testing of different hyperparameters during each test, resulting in the detection of several irregularities in the code and subset dataset. Due to the subsets being chosen randomly for the initial tests there were fewer papers related to some topics which resulted in underfitting for some topics. As a result, for the next test, subsets with the same proportions of each topic as in the initial dataset were used for testing. Following this test, it was decided not to use the entire LitCovid database for fine-tuning due to the incomparable ratios of label topics in the dataset. For example, almost half of the papers were related to one topic, and using the entire dataset caused overfitting, which will be discussed further in the Results section. To address this issue, research papers were selected in similar amounts for each topic, although topics with more papers than others still had more papers in the subset dataset than other topics.

For vector representation and tokenization, BERT's own tokenizer method was used, and the Cross Entropy loss function was used as the main loss function. Cross-validation was also performed during the training process. As BERT takes a long time to fine-tune, checkpoints were saved in case the fine-tuning process stopped unexpectedly, allowing for the process to continue from this checkpoint later.

Eventually, only the best model was saved based on the f1-score metric. After many tests, it was decided to set the batch size to 32, warmup steps to 200, weight decay to 0.01, and number of epochs to 3. Following the fine-tuning and saving of the fine-tuned BERT model, it was tested on the test dataset.

3.6.4 SciBERT and BioBERT

Eventually, SciBERT and BioBERT transformer models were tested. Same as with BERT, these two models required a lot of time and computational power for fine-tuning. Computers with NVIDIA RTX A5000 GPUs (24GB GPU memory) were also used for SciBERT and BioBERT.

Like BERT, both models had vector and tokenizer methods included inside the models. Cross-validation evaluation was performed, models' checkpoints were saved after some period of training time to avoid the loss of training progress and only the best model was kept. The hyperparameter values were given the same values, as they proved to be successful in BERT – 32 to batch size, 200 to warmup steps, 0.01 to weight decay, 3 to epochs amount. Both models then were tested after fine-tuning finished.

3.7 Clusterization

Clusterization involves grouping of documents based on their similarity in feature space. This process allows for the identification of underlying structures or relationships that might not be apparent at first glance. Clustering techniques are valuable in various fields, including biology and medicine, as they facilitate the understanding of complex datasets by simplifying their representation.

In the context of this study, it is expected that clusterization can help to uncover major research themes within the COVID-19 literature. The goal was to find out whether clusterization techniques can help to divide the dataset into different clusters. Two prominent clustering algorithms, K-Means and DBSCAN, were selected for this analysis due to their unique properties and widespread use in various applications.

3.7.1 K-Means

K-Means algorithm requires the value for the number of clusters to be given. As for other techniques, seven clusters were choosing for this algorithm. All three of the vectorization methods were used for K-Means.

As the outcome of K-Means hugely depends on the randomly chosen point at the beginning of the training, it was important to use this method several times to get the best result. For this reason, K-Means was run 50 times and the best trained model was chosen. Silhouette score was used to define the best model, which measures how well each document fits into its assigned cluster based on its proximity to other documents in the same cluster compared to the proximity of the points in the nearest neighboring cluster.

After each research paper was assigned to a cluster, the results were compared using a cross-tabulation matrix, that shows how many papers of certain topic were assigned certain clusters. This can give us an insight on how well the clustering works.

3.8 Tools used

For this research, Python was used as main programming language. According to research by Berkeley University, Python is one of the second most popular programming language in the world [25] and is a great tool for Data Science. Jupyter Notebook was used as a tool to write and program in Python. The codes used for this research can be found in the Appendix and in my GitHub account whose link can also be found in Appendix A.

The most prominent libraries used during Thesis research were pandas, numpy, HuggingFace, scikit-learn. HuggingFace library has useful functions that allowed to use BERT, SciBERT transformers. Scikit-learn contained many useful machine learning algorithms including MNB, LDA, SVM and functions for evaluation of all the used algorithms.

4. RESULTS AND DISCUSSION

In this section, the results and key findings of the Thesis research are presented, detailing the performance of the different classification models, as well as the resulting cluster structure that emerged from the application of clustering and topic modelling techniques. Then the results are discussed and compared in the context of the research objectives, interpreting their significance.

4.1 Classification results

First, the results of supervised machine learning methods are described. Some tables and figure are showing the topic labels as integers, due to BERT models accepting only integers as labels, and for comparison purposes they were kept. The mapping of each topic to integer can be found in Table 1.

4.1.1 Multinomial Naïve Bayes

Multinomial Naïve Bayes algorithm proved itself being as a very fast algorithm and in fact was the fastest classification method out of all used during this research. Both TF-IDF and Count Vectorizer were used as vectorization techniques. However, TF-IDF vectorizer behaved itself very poorly for MNB, despite of trying different methods to improve the results, including putting thresholds on document frequencies of the terms. Only 69% of accuracy, 53% of macro f1-score. On the other hand, CountVectorizer worked quite well for such simple and fast algorithm. MNB with CountVectorizer achieved the accuracy of 86%, the other evaluation metrics are presented in Table 2. In Figure 13, confusion matrix of MNB can be found. The numbers that are shown diagonally from top-left to right-bottom are the amount of correctly predicted labels. The most "confused" topics were Case Report and Diagnosis, which is a consequence of the fact that diagnosis and case reports of COVID-19 infections are often mentioned together.

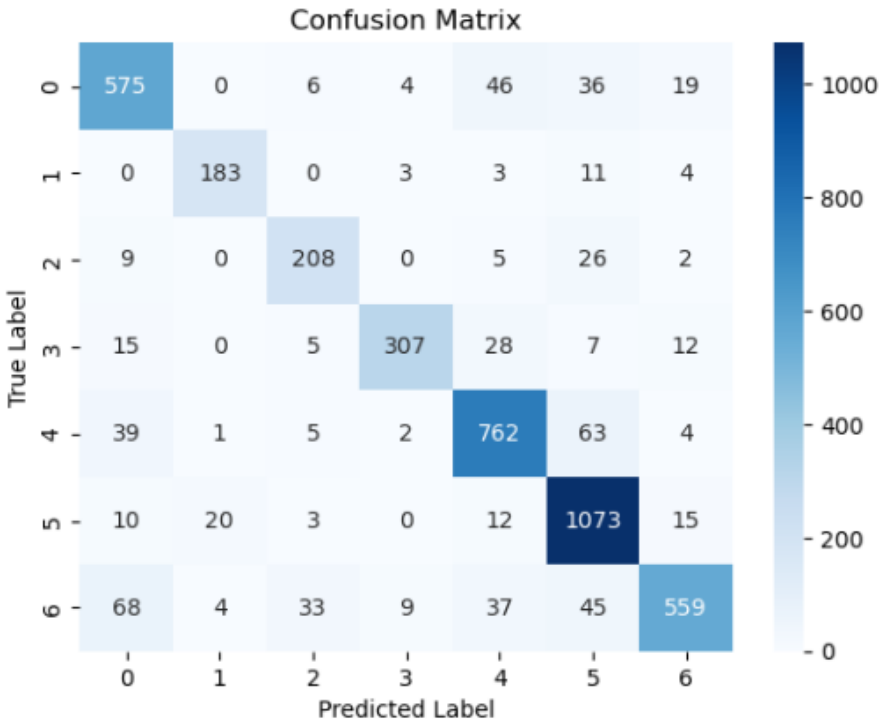


Figure 13. Confusion matrix for MNB

Table 2: Multinomial Naive Bayes classification report

	precision	recall	f1-score
Case Report	0.80	0.84	0.82
Epidemic Forecasting	0.88	0.90	0.89

Long Covid	0.80	0.83	0.82
Mechanism	0.94	0.82	0.88
Treatment	0.85	0.87	0.86
Prevention	0.85	0.95	0.90
Diagnosis	0.91	0.74	0.82
Micro avg	0.86	0.86	0.86
Macro avg	0.86	0.85	0.85
Weighted avg	0.86	0.86	0.86

4.1.2 Support Vector Machines

SVM took a bit longer to train that MNB. All the vectorizer methods mentioned in the methodology section were used for this algorithm. SVM performed itself Count Vectorizer, TF-IDF Vectorizer and Word2Vec very well, though Word2Vec was the most accurate vectorization technique that was used with SVM during this research scoring 90% of accuracy. Table 3 shows the classification report of SVM model and in Figure 14 the confusion matrix can be found.

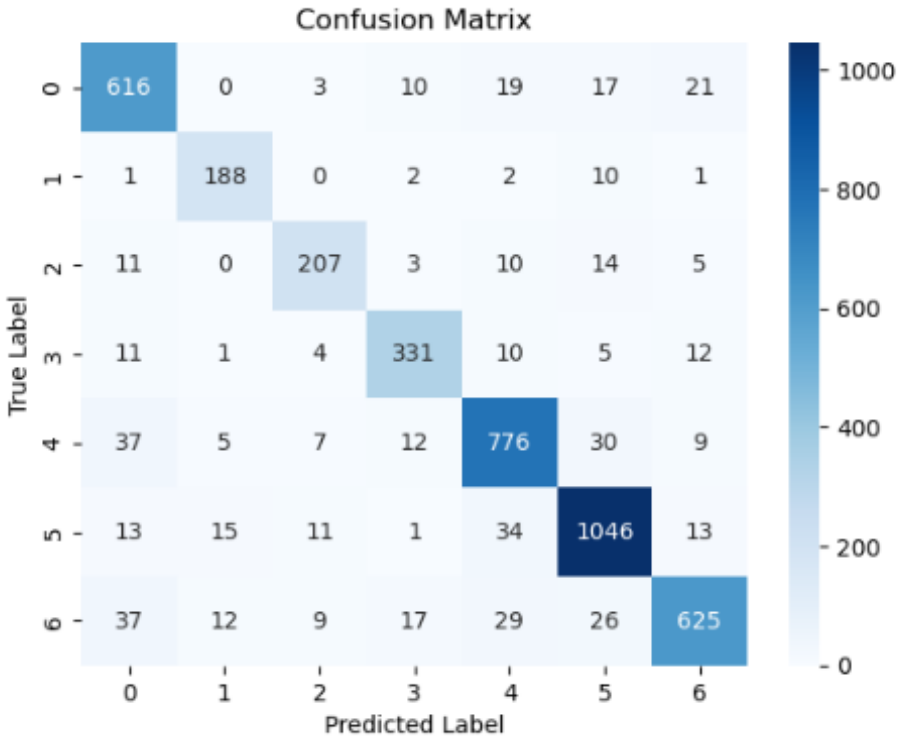


Figure 14. Confusion Matrices for SVM

Table 3: Support Vector Machines classification report

	precision	recall	f1-score
Case Report	0.85	0.89	0.87
Epidemic Forecasting	0.95	0.89	0.92
Long Covid	0.85	0.88	0.87
Mechanism	0.93	0.91	0.92
Treatment	0.91	0.90	0.90

Prevention	0.90	0.95	0.93
Diagnosis	0.93	0.85	0.89
Micro avg	0.90	0.90	0.90
Macro avg	0.90	0.90	0.90
Weighted avg	0.90	0.90	0.90

4.1.3 BERT, SciBERT and BioBERT

All these three models were very slow to fine-tune, sometimes the fine-tuning was taking at least 3 hours, however as the results showed the wait was worth it and the models showed themselves being very accurate. Initial experiments with different hyperparameters were done on BERT and after finding the best values were also applied to SciBERT and BioBERT.

As mentioned earlier, originally the whole dataset was trained on BERT and because of "Prevention" topic accounting for almost half of all the research papers, model overfitted. After making changes to the dataset size, BERT fine-tuning was successful. Same parameters were applied to SciBERT and BioBERT which also showed great results. In fact, all of the three models got 98% accuracy score, which were the most for the supervised algorithms that have been used during this Thesis research. Classification reports are shared in Table 4 for BERT, SciBERT and BioBERT and confusion matrices can be found in Figures 15, 16 and 17.

Table 4: BERT, SciBERT and BioBERT classification reports

	BERT			SciBERT			BioBERT		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Case Report	0.97	0.99	0.98	0.97	0.98	0.98	0.97	0.98	0.98
Epidemic Forecasting	0.98	0.98	0.98	0.96	0.99	0.97	0.98	0.98	0.98
Long Covid	0.98	0.97	0.97	0.99	0.96	0.97	0.98	0.96	0.97
Mechanism	0.99	0.98	0.99	0.97	0.99	0.98	0.99	0.98	0.98
Treatment	0.97	0.99	0.98	0.98	0.99	0.99	0.98	0.98	0.98
Prevention	0.99	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Diagnosis	0.98	0.97	0.98	0.98	0.97	0.98	0.98	0.98	0.98
Micro avg	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Macro avg	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Weighted avg	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98

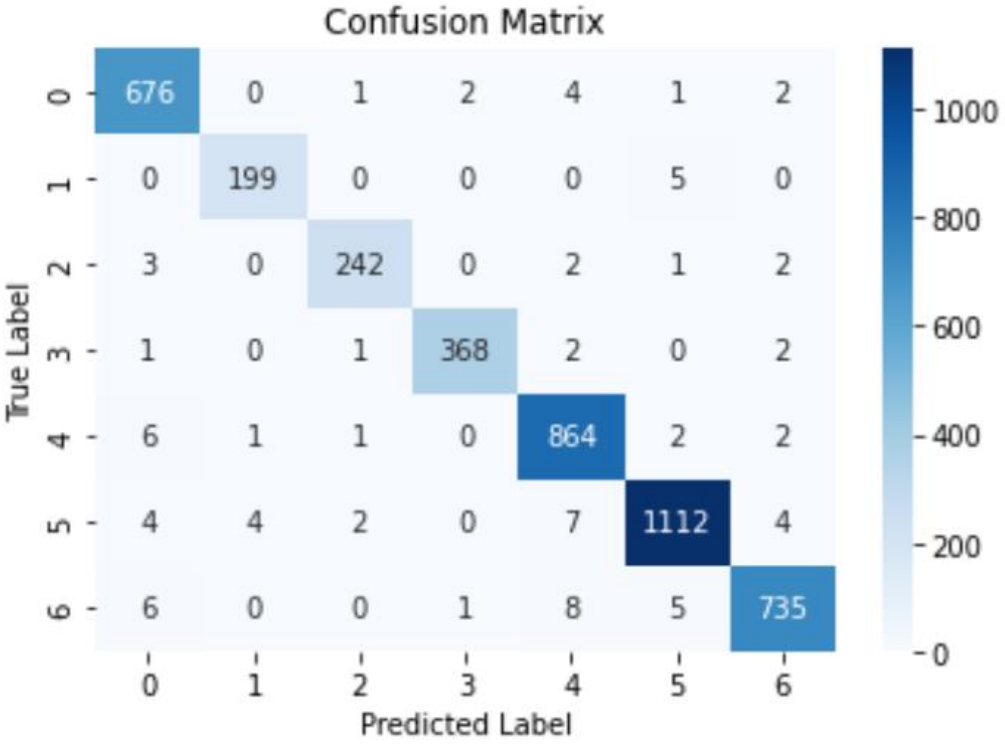


Figure 15. BERT confusion matrix

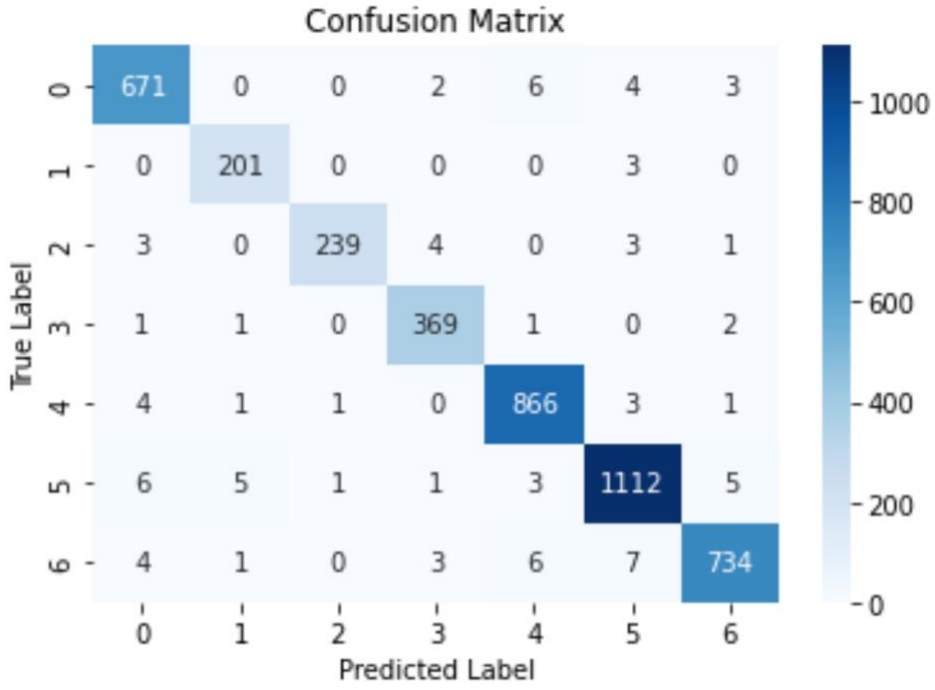


Figure 16. SciBERT confusion matrix

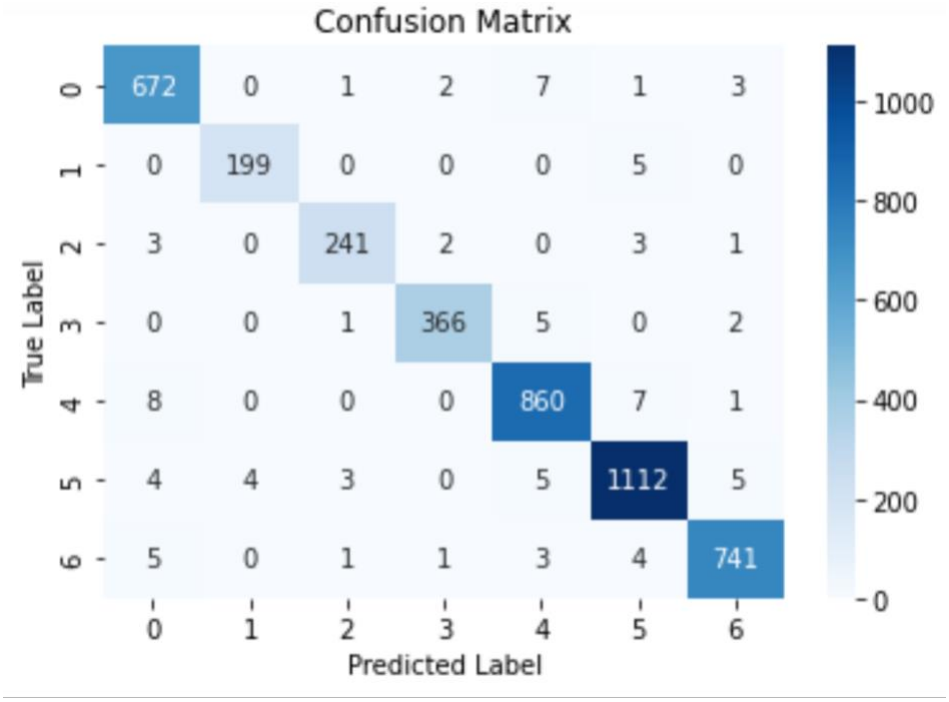


Figure 17. BioBERT confusion matrix

4.2 Topic Modelling and Clustering

This section is devoted to the results of unsupervised machine learning algorithms. First, Topic Modelling results are discussed and then the results of Clusterization algorithms are shown.

4.2.1 Latent Dirichlet Allocation

Before applying LDA for LitCovid, the author has tested it on CORD-19 dataset. LDA was able to define several topics. The expectation was that LDA will give similar results for LitCovid dataset, too. Nevertheless, the results were unexpected. As can be seen on Figure 18, LDA derived mostly only one topic, with only six papers being assigned to other topics. While this result was not expected one, diving deep into LDA algorithm allowed to find out the cause of these results. The LitCovid used for this Thesis contain research papers with very similar content – COVID-19, making it difficult for the LDA model to identify distinct topics. When the documents in a dataset share a high degree of similarity, the LDA algorithm can have difficulty in distinguishing separate topics. The topics derived by LDA can be found in Table 5.

Table 5: LDA Topics and ten most frequent words in each topic

Topic #0	Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6
greenness	dect	febridx	igan	covid	agvhd	seraph
microvilli	ssnhl	apeced	ittp	19	ltr	mbc
hbcr	cper	ensitrelvir	aird	patient	dhp	rvu
teleglaucoma	hcc	nimotuzumab	cov2ag	sars	scribe	iepo
cd47	spikogen	aecopd	caplacizumab	cov	dfm	cehc
kovir	hbcu	psma	pal	study	m1273	lipschutz
khosta	teleaudiology	melioidosis	bppv	vaccine	asahii	microbind
splashguard	noq19	hinzii	tapp	case	pwcf	smartamp
pyroglutamine	enpatoran	merv	rcfr	infection	denture	gwpr
mmym	iief	vitd	prca	disease	drr	cov1901

4.2.2 Non-negative Matrix Factorization

Once the best performing NMF model with the highest coherence score of 0.6424 was identified, the seven topics of the model were printed. Unlike the LDA model, NMF was able to extract meaningful topics from the LitCovid dataset. Table 6 displays the topics identified by NMF and the ten most frequent words associated with each topic. By examining these words, it becomes possible to gain insight into the content of each topic. For example, Topic #0 appears to be about COVID-19 itself, Topic #1 is focused on vaccination, Topic #2 deals with SARS-CoV-2 virus mutations and variations, Topic #3 is likely about anxiety related to COVID-19 and the pandemic, Topic #4 is focused on pandemic predictions, Topic #5 is about COVID-19 tests, and Topic #6 is most likely concerned with the COVID-19 pandemic in general.

When comparing these topics to the results of the cross-tabulation matrix shown in Figure 20, interesting correlations between the Topics identified by NMF and the actual topics can be observed. For instance, Topic #1 is about vaccination, which is a method of treating COVID-19. Similarly, Topic #5 is related to COVID-19 tests, which are crucial for detecting and diagnosing the virus.

Table 6: MNF Topics and ten most frequent words in each topic

Topic #0	Topic #1	Topic #2	Topic #3	Topic #4	Topic #5	Topic #6
patient	vaccine	cov	study	model	test	health
case	vaccination	sars	symptom	case	cov	care
treatment	dose	protein	group	datum	sars	pandemic
disease	mrna	variant	participant	image	rt	service
clinical	vaccinate	virus	anxiety	learning	sample	healthcare
severe	bnt162b2	cell	1	number	assay	public
respiratory	antibody	mutation	child	prediction	pcr	telemedicine
infection	booster	sequence	depression	propose	positive	social
acute	response	genome	post	epidemic	detection	patient
report	receive	spike	score	forecast	antibody	telehealth

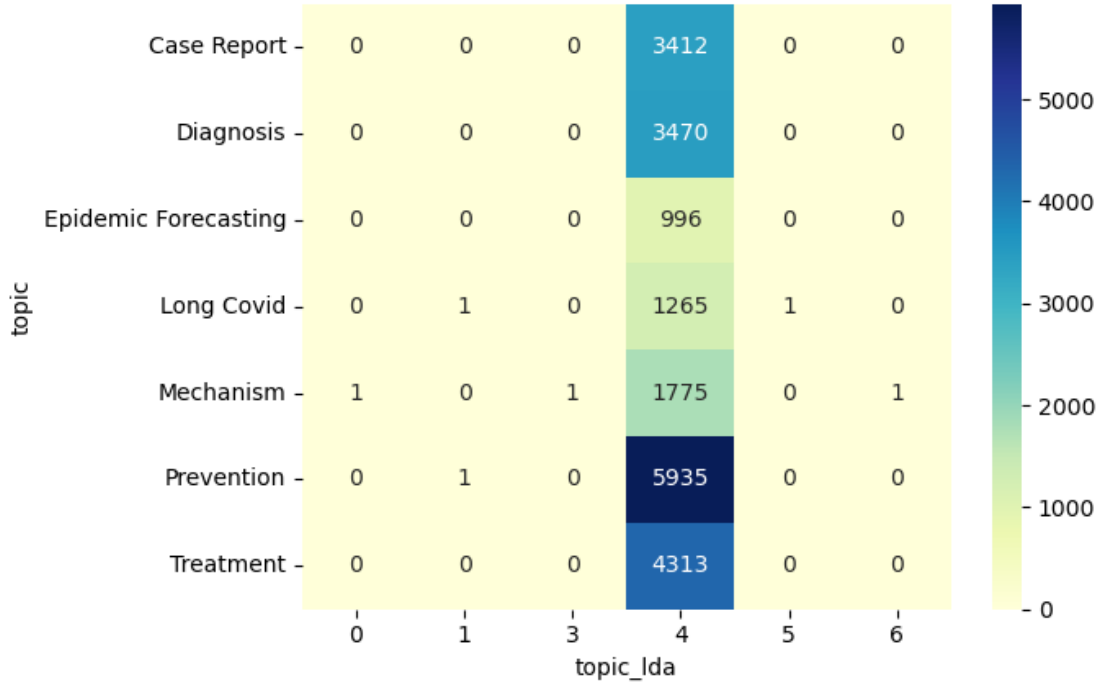


Figure 18. Cross-tabulation matrices of LDA assigned topics with real topics.

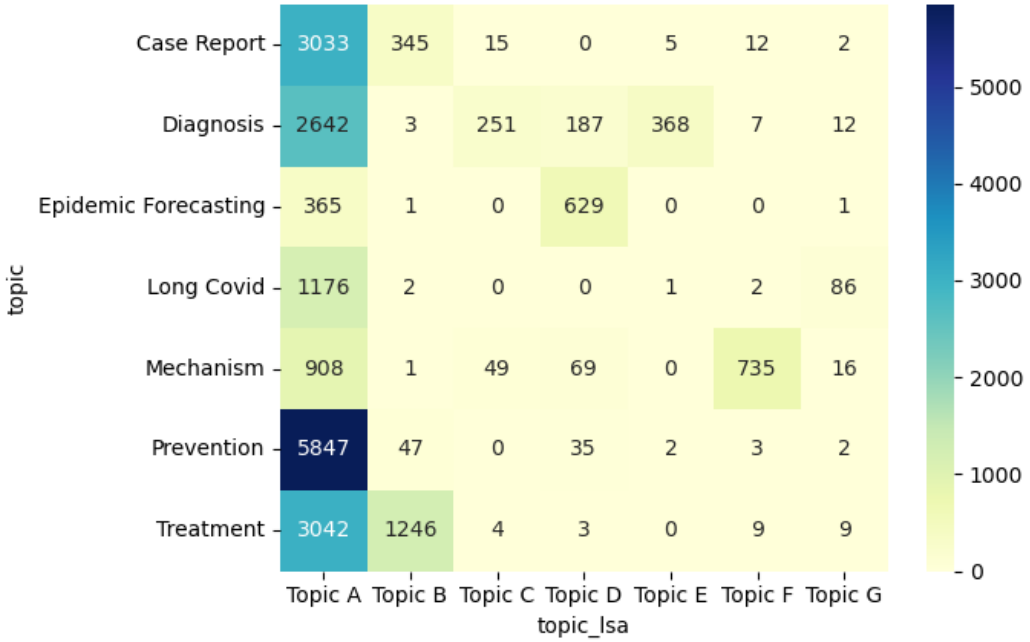


Figure 19. Cross-tabulation matrices of LSA assigned topics with real topics.

4.2.3 Latent Semantic Analysis

The last topic modeling technique utilized in the thesis research was Latent Semantic Analysis (LSA). This model demonstrated fast computational performance and successfully extracted seven topics, each with ten most frequent words, as shown in Table 7. The results of LSA outperformed those of Latent Dirichlet Allocation (LDA) in terms of topic extraction. However, according to the cross-tabulation matrix presented in Figure 19, the assignment of topics to the research papers using LSA was not as accurate as with Non-Negative Matrix Factorization (NMF).

Table 7: LSA Topics and ten most frequent words in each topic

Topic A	Topic B	Topic C	Topic D	Topic E	Topic F	Topic G
vaccine	vaccine	antibody	model	test	mutation	symptom
health	vaccination	protein	variant	rt	variant	variant
vaccination	dose	cell	sample	sample	protein	group
test	mrna	assay	mutation	assay	cell	mutation
clinical	vaccinate	rt	sequence	pcr	sequence	omicron
symptom	bnt162b2	sample	prediction	positive	virus	1
2020	antibody	viral	detection	antibody	genome	anxiety
model	booster	virus	number	health	spike	depression
care	hesitancy	test	learning	testing	health	child
datum	pfizer	pcr	datum	detection	ace2	2020

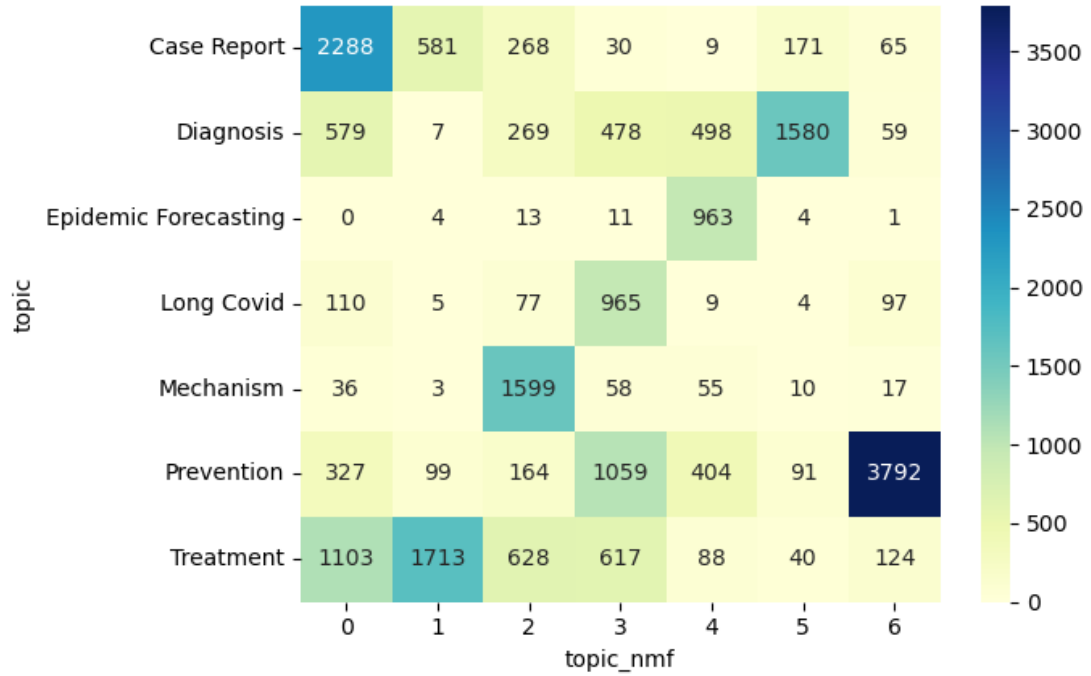


Figure 20. Cross-tabulation matrices of NMF with real topics.

4.2.4 K-Means

As outlined in the methodology section, the K-Means algorithm was utilized in this study, and the model was trained a total of fifty times to the best clustering solution was chosen to work with. The silhouette score of the best-performing K-Means model was found to be 0.00994, indicating that the clustering solution is only marginally better than a random allocation of data points. The dataset was ultimately divided into seven clusters using the optimal K-Means model.

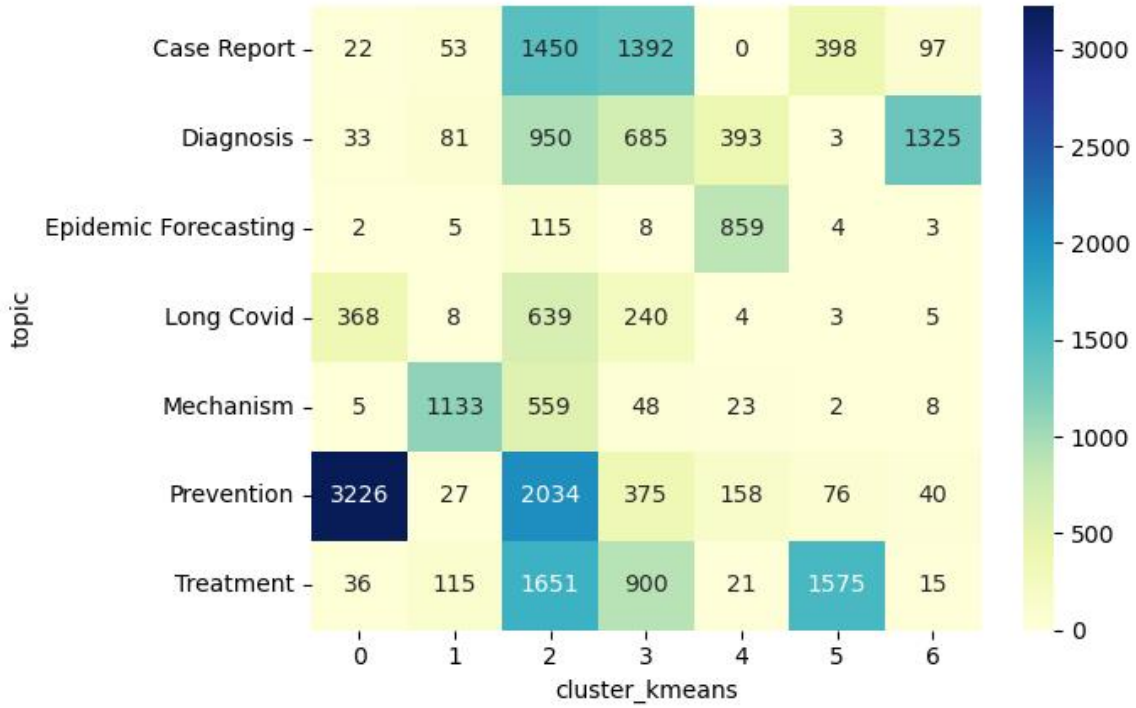


Figure 21. Cross-tabulation matrices of K-Means with real topics.

Figure 21 displays a cross-tabulation matrix of the clusters with real topics, providing insight into potential correlations between the clusters and topics. However, the relationship between certain topics

and clusters is not as evident in K-Means compared to NMF. Specifically, cluster 0 consists primarily of publications related to Prevention, cluster 1 is associated with Mechanism, cluster 5 with Treatment, and cluster 6 with Diagnosis. Nonetheless, other clusters encompass numerous publications from various topics, which makes it difficult to establish a clear association between those topics and the respective clusters.

4.3 Comparison

In this chapter the results of different supervised and unsupervised machine learning algorithms were shown and discussed.

Among the classification algorithms, MNB was the fastest algorithm. It performed better when using CountVectorizer than with TF-IDF Vectorizer, while SVM performed better with Word2Vec. BERT, SciBERT, and BioBERT models were the slowest to fine-tune, but they demonstrated the highest accuracy of 98% among all the classification models. These three models outperformed both MNB and SVM, making them the most accurate supervised learning algorithms used in the thesis research.

LDA performed poorly on the LitCovid dataset, deriving mostly one dominant topic. On the other hand, NMF was more successful in extracting meaningful topics from the dataset. LSA was the fastest unsupervised algorithm, however, it was not as accurate in assigning topics to research papers and deriving topics from the dataset as NMF. In K-Means, while some clusters demonstrated strong associations with specific topics, other clusters consisted of mixed publications, making it difficult to establish clear relationships.

In conclusion, BERT, SciBERT, and BioBERT were the top-performing classification models, while NMF outperformed other unsupervised learning techniques in extracting meaningful topics from the dataset.

5 CONCLUSION AND FUTURE WORKS

In this study, the author shared his investigations on various techniques for text classification and clustering applied to a dataset of COVID-19 research papers obtained from the LitCovid database. The main findings revealed that the BERT, SciBERT and BioBERT transformer models were highly effective for text classification tasks, outperforming traditional machine learning methods. This success can be attributed to BERT's ability to capture bidirectional context, leverage pre-trained knowledge from a large corpus, and adapt to specific tasks through fine-tuning.

The comparison of classification and topic modelling results provided valuable insights into the relationships and structure of the research papers. While classification techniques allowed us to assign predefined labels to the documents, topic modelling helped to uncover latent patterns and groupings within the data that may not have been apparent otherwise. NMF topic modelling technique was especially useful in achieving this. The combination of these two approaches enhances the understanding of the COVID-19 related literature, enabling a more comprehensive analysis.

There are several possible directions for future work. First, the classification models could be further improved by incorporating additional pre-processing techniques, such as domain-specific tokenization or customized embeddings trained on scientific literature. Combining BERT, SciBERT or BioBERT with NMF may allow us to find even more hidden insights in COVID-19 literature. This combination may also be applied to unlabeled CORD-19 dataset, which has three times more research papers than LitCovid.

In conclusion, this study demonstrated the effectiveness of combining classification and clustering techniques in analyzing the COVID-19 literature. The insights gained from this work not only contribute to a better understanding of the current state of COVID-19 research but also provide a foundation for further investigation and application of these methods in other scientific domains.

REFERENCES

[1] Lu Wang L, Lo K, Chandrasekhar Y, Reas R, Yang J, Eide D, Funk K, Kinney R, Liu Z, Merrill W, Mooney P, Murdick D, Rishi D, Sheehan J, Shen Z, Stilson B, Wade AD, Wang K, Wilhelm C, Xie B, Raymond D, Weld DS, Etzioni O, Kohlmeier S. CORD-19: The Covid-19 Open Research Dataset. ArXiv [Preprint]. 2020 Apr 22:arXiv:2004.10706v2. PMID: 32510522; PMCID: PMC7251955.

[2] @book{Goodfellow-et-al-2016, title={Deep Learning}, author={Ian Goodfellow and Yoshua Bengio and Aaron Courville}, publisher={MIT Press}, note={\url{http://www.deeplearningbook.org}}, year={2016}

[3] The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) 2nd Edition by Hastie, Tibshirani, Friedman

[4] Scheinfeld. (2022). LitCovid. Journal of the Medical Library Association, 110(2), 279–. https://doi.org/10.5195/jmla.2022.1274

[5] Chen, Allot, A., & Lu, Z. (2021). LitCovid: an open database of COVID-19 literature. Nucleic Acids Research, 49(D1), D1534–D1540. https://doi.org/10.1093/nar/gkaa952

[6] Chen, Allot, A., Leaman, R., Wei, C.-H., Aghaarabi, E., Guerrerio, J. J., Xu, L., & Lu, Z. (2023). LitCovid in 2022: an information resource for the COVID-19 literature. Nucleic Acids Research, 51(D1), D1512–D1518. https://doi.org/10.1093/nar/gkac1005

[7] https://www.ncbi.nlm.nih.gov/research/coronavirus/

[8] Dastani, & Danesh, F. (2021). Iranian COVID-19 Publications in LitCovid: Text Mining and Topic Modeling. Scientific Programming, 2021, 1–12. https://doi.org/10.1155/2021/3315695

[9] Khateeb, Ahmad Arsalan. "BERT Models in Document Classification." Vysoká škola ekonomická v Praze, 2022. Print.

[10] David M. Blei. 2012. Probabilistic topic models. Commun. ACM 55, 4 (April 2012), 77–84. https://doi.org/10.1145/2133806.2133826

[11] Sperandeo R, Messina G, Iennaco D, Sessa F, Russo V, Polito R, Monda V, Monda M, Messina A, Mosca LL, Mosca L, Dell'Orco S, Moretto E, Gigante E, Chiacchio A, Scognamiglio C, Carotenuto M and Maldonato NM (2020) What Does Personality Mean in the Context of Mental Health? A Topic Modeling Approach Based on Abstracts Published in Pubmed Over the Last 5 Years. Front. Psychiatry 10:938. doi: 10.3389/fpsyt.2019.00938

[12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. J. Mach. Learn. Res. 3, null (3/1/2003), 993–1022.

[13] https://www.mbmlbook.com/ModelAnalysis_Latent_Dirichlet_Allocation.html

[14] Boyack KW, Newman D, Duhon RJ, Klavans R, Patek M, Biberstine JR, Schijvenaars B, Skupin A, Ma N, Börner K. Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. PLoS One. 2011 Mar 17;6(3):e18029. doi: 10.1371/journal.pone.0018029. PMID: 21437291; PMCID: PMC3060097.

[15] Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.
[17] Zulfany Erlisa Rasjid, Reina Setiawan, Performance Comparison and Optimization of Text Document Classification using k-NN and Naïve Bayes Classification Techniques, Procedia Computer Science, Volume 116, 2017, Pages 107-112, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.10.017
[18] B. Y. Pratama and R. Sarno, "Personality classification based on Twitter text using Naive Bayes, KNN and SVM," 2015 International Conference on Data and Software Engineering (ICoDSE), Yogyakarta, Indonesia, 2015, pp. 170-174, doi: 10.1109/ICODSE.2015.7436992.
[19] Ramya, B.N., Shetty, S.M., Amaresh, A.M. et al. Smart Simon Bot with Public Sentiment Analysis for Novel Covid-19 Tweets Stratification. SN COMPUT. SCI. 2, 227 (2021), pp. 1–11. https://doi.org/10.1007/s42979-021-00625-5
[20] https://allenai.github.io/scispacy/
[21] Colavizza G, Costas R, Traag VA, van Eck NJ, van Leeuwen T, Waltman L (2021) A scientometric overview of CORD-19. PLoS ONE 16(1): e0244839. https://doi.org/10.1371/journal.pone.0244839
[22] X. Hu and R. Zhang, "Text classification based on machine learning," *2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, 2022, pp. 911-916, doi: 10.1109/ICAICA54878.2022.9844556.
[23] Krzeszewska, U.; Poniszewska-Mara ´nda, A.; Ochelska-Mierzejewska, J. Systematic Comparison of Vectorization Methods in Classification Context. Appl. Sci. 2022, 12, 5119. https:// doi.org/10.3390/app12105119
[24] W. Tian, J. Li and H. Li, "A Method of Feature Selection Based on Word2Vec in Text Categorization," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 9452-9455, doi: 10.23919/ChiCC.2018.8483345.
[25] 11 Most In-Demand Programming Languages in 2023 - Berkeley Boot Camps. (2020, December 16). Berkeley Boot Camps. https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/
[26] SciBERT: A Pretrained Language Model for Scientific Text (https://aclanthology.org/D19-1371) (Beltagy et al., EMNLP-IJCNLP 2019)
[27] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020 Feb 15;36(4):1234-1240. doi: 10.1093/bioinformatics/btz682. PMID: 31501885; PMCID: PMC7703786.
[28] Advances in Communication Technology, Computing and Engineering Editors: Mariyam Ouaissa, Mariya Ouaissa, Sarah El Himer, Zakaria Boulouard pp. 122 – 133, Copyright © 2021 RGN Publications COVID-19 Tweets Sentiment Analysis using Machine Learning Approaches and Divers Document Representations https://rgnpublications.com/ICACTCE2021/manuscripts/012-83.pdf
[29] B. Ramesh, J.G.R. Sathiaseelan, An Advanced Multi Class Instance Selection based Support Vector Machine for Text Classification, Procedia Computer Science, Volume 57, 2015, Pages 1124-1130, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.07.400.
[30] Support Vector Machines: Data Analysis, Machine Learning and Applications: Data Analysis, Machine Learning and Applications, edited by Brandon H. Boyle, Nova Science Publishers, Incorporated, 2011. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/adaaz/detail.action?docID=3021500.
[31] Choubey, V. (2020, July 21). Topic Modelling Using NMF. Medium. https://medium.com/voice-tech-podcast/topic-modelling-using-nmf-2f510d962b6e
[32] Valdez, D., Pickett, A.C. and Goodson, P. (2018), Topic Modeling: Latent Semantic Analysis for the Social Sciences. Social Science Quarterly, 99: 1665-1679. https://doi-org.proxygw.wrlc.org/10.1111/ssqu.12528
[33] [1] Karim, Rezaul. TensorFlow: Powerful Predictive Analytics with TensorFlow : Predict Valuable Insights of Your Data with TensorFlow, Packt Publishing, Limited, 2018. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/adaaz/detail.action?docID=5322213.
[34] [2] Kotu, Vijay, and Bala Deshpande. Predictive Analytics and Data Mining : Concepts and Practice with RapidMiner, Elsevier Science & Technology, 2014. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/adaaz/detail.action?docID=1875324.

APPENDICES

A.1 GitHub repository

To access the codes written and used during this Thesis study, follow this link:

- https://github.com/rustam-007/MasterThesis

The repository has code snippets for reading JSON file of LitCovid database. All the algorithms mentioned in the main part of Thesis are also posted in this GitHub repository.