FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

TEXT CLUSTERING AND TOPIC MODELING ON COVID-19 VACCINE TWEETS USING MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, AND DEEP LEARNING

David Okore UKWEN

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

OCTOBER 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

TEXT CLUSTERING AND TOPIC MODELING ON COVID-19 VACCINE TWEETS USING MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, AND DEEP LEARNING

Author
David Okore UKWEN

Supervisor
Prof. Dr. Murat KARABATAK

OCTOBER 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

Title:	Text Clustering and Topic Modeling on COVID-19 Vaccine Tweets Using Machine Learning, Natural Language Processing, and Deep Learning
Author:	David Okore UKWEN
Submission Date:	17 September 2022
Defense Date:	24 October 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Prof. Dr. Murat KARABATAK Firat University, Faculty of Technology	Approved
Chair:	Assoc. Prof. Dr. Özal YILDIRIM Fırat University, Faculty of Technology	Approved
Member:	Assist. Prof. Dr. Ahmet Arif AYDIN Inonu University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Burhan ERGEN
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “Text Clustering and Topic Modeling on COVID-19 Vaccine Tweets Using Machine Learning, Natural Language Processing, and Deep Learning” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

24 October 2022

David Okore UKWEN

PREFACE

We live in a time where quarantine, wearing of masks, and social distancing are the norm. This is a result of the coronavirus pandemic which has disrupted the way we live our daily lives. Several vaccination programs are going on presently against the novel coronavirus. Topic modeling and Text clustering are the easiest means of getting insights from data without labels. This is necessary as the number of observations is in the hundreds of millions and any attempt to label the data will cost enormous money and time, thus making it difficult to understand the data quickly. Deep learning and Machine learning techniques have proven to be the go-to artificial intelligence algorithms to obtain insight from natural language data due to their high level of precision and accuracy.

In this study, The concept of text clustering and topic modeling are introduced, various clustering and topic modeling algorithms are discussed, and dimensionality reduction techniques are presented. Deep learning, topic modeling, and machine learning algorithms are combined to cluster coronavirus COVID-19 vaccine tweets sourced from Twitter and as well extract relevant topics prevalent in the corpus, which reflects the ongoing trends in the vaccination efforts.

I send my deepest appreciation, gratitude, respect, and thanksgiving to the National Information Technology Development Agency (NITDA) for sponsoring my master's degree program. I will also not fail to express my appreciation to Assoc. Dr. Murat Karabatak and Assoc. Dr. Ozal Yildrim for their immense effort and input to see that this thesis was a success. I also send my appreciation to all the professors and lecturers in the Department of Software Engineering at Fırat University for their contributions.

David Okore UKWEN
ELAZIG, 2022

TABLE OF CONTENTS

Page

ABSTRACT

Text Clustering and Topic Modeling on COVID-19 Vaccine Tweets Using Machine Learning, Natural Language Processing, and Deep Learning

David Okore UKWEN

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

October 2022, Page: xiii + 136

In late December 2019, the unique coronavirus disease (COVID-19) emerged, causing a tremendous loss of life throughout the globe and posing a previously unheard-of challenge to public health, education, social life, global economies, and the workplace. To end the COVID-19 pandemic, equitable access to safe and effective vaccinations is essential. The best approaches to quickly gain an understanding of COVID-19 text data presented in the literature are those that use unsupervised learning.

The goal of this research thesis is to use text clustering and topic modeling to analyze coronavirus vaccine tweets. Using machine learning and deep learning methods and techniques, it investigates the optimal number of topics and clusters prevalent in the coronavirus (COVID-19) vaccine corpus. The study also looks into various insights that can be extracted from tweets through exploratory data analysis, using word embeddings to improve the accuracy of the proposed models, evaluate unsupervised learning methods, and gain other insights.

Text clustering was performed using machine learning clustering techniques and algorithms like k-means and HDBSCAN, deep learning-based clustering methods, and dimensionality reduction algorithms such as PCA, LDA, t-SNE, and UMAP, while topic modeling algorithms such as LDA, GSDMM, and TopicBERT/ BERTopic were used to obtain relevant topics from the coronavirus vaccine corpora.

The findings of this study demonstrated that GSDMM and BERTopic produced significant topics from the COVID-19 corpus, while deep learning clustering methods outperformed their machine learning counterparts in text clustering. K-means performed superior clustering based on multiple assessment criteria, but HDBSCAN performed better clustering based on features learned.

Keywords: Text clustering, Machine learning, Topic Modeling, Deep learning, Dimensionality reduction

ÖZET

Makine Öğrenimi, Doğal Dil İşleme ve Derin Öğrenme Kullanılarak COVID-19 Aşısı Tweetlerinde Metin Kümeleme ve Konu modelleme

David Okore UKWEN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Ekim 2022, Sayfa: xiii + 136

Korona virüs hastalığı (COVID-19), Aralık 2019'un sonlarında ortaya çıkmış ve dünya genelinde muazzam bir can kaybına neden olmuş, halk sağlığı, eğitim, sosyal yaşam, küresel ekonomiler ve iş yeri için çok büyük sorunlar ortaya çıkarmıştır. COVID-19 pandemisini sona erdirmek için güvenli ve etkili aşılara erişim gerekmektedir. Literatürde bulunan COVID-19 metin verilerini hızlı bir şekilde yorumlamak için kullanılan en iyi yaklaşımlar genellikle denetimsiz öğrenme yaklaşımları olmuştur.

Bu tezinin amacı, korona virüs aşısı ile ilgili tweet'leri analiz etmek için metinlerin kümelemesi ve konuların modellemeyi kullanmaktır. Makine öğrenmesi ve derin öğrenme yöntem ve tekniklerini kullanarak, korona virüs (COVID-19) aşısı ile ilgili yaygın olan konular ve kümeler araştırılmıştır. Çalışma ayrıca, önerilen modellerin doğruluğunu artırmak, denetimsiz öğrenme yöntemlerini değerlendirmek ve başka bulgular elde etmek için kelime yerleştirmelerini kullanarak, keşifsel veri analizi yoluyla tweet'lerden çıkarılabilecek çeşitli sonuçları da incelemektedir.

Tezde, metin kümeleme, k-means ve HDBSCAN gibi makine öğrenmesi kümeleme algoritmaları, derin öğrenme tabanlı kümeleme yöntemleri ve PCA, LDA, t-SNE ve UMAP gibi boyut azaltma algoritmaları kullanılmış ayrıca, LDA gibi konu modelleme algoritmaları ile GSDMM ve TopicBERT/ BERTopic yöntemleri, korona virüs aşısı ile ilgili sonuçları elde etmek için kullanılmıştır.

Bu çalışmanın bulguları, GSDMM ve BERTopic'in COVID-19 ile ilgili önemli başlıkları elde ettiğini, derin öğrenme kümeleme yöntemlerinin metin kümelemede diğer makine öğrenmesi yöntemlerinden daha iyi performans gösterdiğini ortaya çıkarmıştır. K-ortalamalar yöntemi, çoklu değerlendirme kriterlerine dayalı olarak başarılı bir kümeleme gerçekleştirmiş ancak HDBSCAN yöntemi öğrenilen özelliklere dayalı olarak daha iyi kümeleme gerçekleştirdiği ortaya çıkmıştır.

Anahtar Kelimeler: Metin kümeleme, Makine öğrenimi, Konu Modelleme, Derin öğrenme, Boyut azaltma

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF APPENDICES

Page

SYMBOLS AND ABBREVIATIONS

Symbols

β, σ	: Greek Alphabets
P	: Probability

Abbreviations

STC	: Short Text Clustering
DLVN	: Deep Learning Vocabulary Network
MCKSM	: Multi-Cluster Spherical K-Means
LSTM	: Long Short Term Memory
CNN	: Convolutional Neural Networks
BiLSTM	: Bi-directional Long Short Term Memory
VAE	: Variational AutoEncoder
AE	: AutoEncoder
PCA	: Principal Component Analysis
ICA	: Independent Component Analysis
SVD	: Singular Value Decomposition
NMF	: Non-Negative Matrix Factorization
PAM	: Partition Around Medoids
CLARA	: Clustering Large Applications
CLARANS	: Clustering Large Applications based on RANdomized Search
ROCK	: Robust Clustering Algorithm for Categorical Attributes
BIRCH	: Balanced Iterative Reducing and Clustering using Hierarchies
UPGMA	: Unweighted Pair Group Method with Arithmetic mean
DBCLASD	: Distribution-Based Clustering Algorithm for mining in large Spatial DB
OPTICS	: Ordering Points To Identify the Clustering Structure
DENCLUE	: DENsity CLUstEring
HDBSCAN	: Hierarchical Density-Based Spatial Clustering of Applications with Noise
UMAP	: Uniform Manifold Approximation and Projection
RNN	: Recurrent Neural Networks
KLD	: Kullback-Leibler Divergence
MRF	: Random Markov Field
TF-IDF	: Term Frequency and Inverse Document Frequency (TF*IDF)
DNN	: Deep Neural Networks.
EM	: Expectation Maximization.

1. INTRODUCTION

Overview of Study

Textual clustering of coronavirus vaccine tweets involves clustering the COVID-19 tweets into groups based on similarity. Topic modeling on COVID-19 vaccine tweets, on the other hand, seeks to find the abstract salient terms (topics) that occur in the COVID-19 vaccine tweet by detecting and clustering word and phrase patterns within the document. Modern natural language processing methods, deep learning algorithms, and machine learning have made it easier to manage text documents (tweets) by converting text into 0's and 1's that computers can understand, then assign clusters and topics by learning the underlying data representations.

People classify text in different ways. For example, libraries have a genre system, books have chapters, and newspapers have sections for different types of news. However, in most cases, we can suggest different divisions that are effective and valuable for the same set of text. Also, in most cases, different people have different ways of splitting the same amount of text. This is not as a matter of course a bad thing if we accept and understand the reason for implementing it. Each new set of subdivisions of text can offer new ideas [1].

This research aims to identify and evaluate text clustering and topic modeling approaches for COVID-19 vaccine tweets using various advanced natural language processing methods, classical machine-learning clustering, topic modeling techniques, and deep learning-based clustering algorithms in an unsupervised manner to generate clusters and topics within the clusters. This chapter will introduce the study by first discussing the background and context, followed by the research problem, the research aims, objectives, and questions, why the study was significant, and finally, the scope of the research.

Background of Study

Quarantine, wearing a mask, and social distancing have become the norm in recent years. This is a result of the novel COVID-19 disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), disrupting our daily lives. After an outbreak in China in December 2019, WHO – (the World Health Organization) identified SARS-CoV-2 as a new strain of coronavirus and declared it a pandemic in early 2020 as the outbreak spread rapidly around the world. The coronavirus (COVID-19) is a disease caused by SARS-CoV-2 that can cause respiratory infections. The upper respiratory tract (throat, nose, sinuses) or lower respiratory tract (lungs and trachea) may be affected. Like other coronaviruses, the main mechanism of spread is person-to-person contact. Infections range from mild to fatal. Most people who get the virus experience moderate to mild respiratory illness that clears up on their own without specific medical treatment. However, some people get very sick and need treatment. Older adults and those with underlying

medical conditions such as diabetes, heart disease, cancer, or chronic respiratory disease are more likely to develop serious illnesses. SARS-CoV-2 is one of seven coronaviruses, including those that cause serious illnesses such as MERS – (Middle East respiratory syndrome) and SARS – (sudden acute respiratory syndrome). Other coronaviruses that cause most colds affect us throughout the year and do not pose a serious threat to otherwise healthy people. [2, 3].

Currently, many vaccination efforts geared toward combating the coronavirus are underway. At least 246 vaccine projects have started against Covid-19 since January 2020: WHO - (World Health Organization) now counts 329 (list as of January 13, 2022). This number incorporates vaccine projects in clinical and preclinical trials and development. The main COVID-19 vaccines currently used to prevent the spread of the coronavirus diease are Johnson and Johnson, Moderna, Pfizer/BioNTech, Sputnik V, Oxford/AstraZeneca, Sinovac, CanSino, and Sinopharm/Beijing. These vaccines help our bodies fight off coronavirus diseases by stimulating the body to make antibodies. Even if a person is later exposed to the virus, the antibodies still protect against the virus. The availability of each vaccine mainly depends on the recipient's region [4].

Natural Language Processing (NLP) is a research area under artificial intelligence that aims to be able to process, understand, analyze, and transform the natural language produced by humans. NLP is a research field that includes linguistics, computer science, and artificial intelligence. Natural language processing combines insights from statistics, computational linguistics, deep learning, and machine learning to process the human natural language in text or speech data format. Natural language processing allows us to understand the structure and meaning of human natural language by analyzing various aspects of semantics, grammar, morphology, and pragmatics. Computer science then translates language knowledge into rule-based machine learning algorithms that can solve different problems and perform desired tasks.

The early natural language processing systems started as a machine translation (MT) system in the 1950s. The systems were designed for computer-human interaction to aid communication. Since its inception, natural language processing has undergone four important stages of development. These stages are characterized by the need for the use of artificial intelligence, machine translation, the processing of large amounts of linguistic data, and the use of logical grammatical styles [5].

Contrary to popular opinion, it is more laborious to know grammar rules, the meaning of phrases, and metaphorical expressions in any language than to process numerical data. Frequently, a context or foreknowledge on the topic of the text is required. However, in numerical data processing, mathematical expressions are enough to solve problems without any need for further information such as the context. For instance, ordinarily, the computer does not know that water is a liquid form, and ice is a solid form, but with statistical methods, the relationship between the two terms can be established.

Natural language processing is such a need because it enables the analysis of communication tools and the data produced during communication: By 2025, the data that will be produced daily is estimated to be about 463 Exabytes (equivalent to 212.765.957 DVDs). It is expected that the bulk of this data consists of written content or content that can be translated into text (video, audio). In this space, it is not possible to start an activity that can analyze and understand the content created by various people in their language. For this reason, technologies that make this size of data meaningful to analyze at various levels have emerged as a vital need [6].

The main artificial intelligence (AI) technology discussed in the recent literature is NLP – (natural language processing). NLP-based systems find applications in a wide variety of industries: finance, commerce, business, sports, healthcare, digital forensics, recruitment, energy and the environment, cybersecurity, defense, and various automated tasks related to text analysis.

Machine learning is the branch of AI – (artificial intelligence) and computer science, which focuses on the use of data and algorithms to learn and gradually improve accuracy without the need for explicit programming. With the emergence of new computer technologies, machine learning today is different from the machine learning practiced in the past. It comes from design acknowledgment and the hypothesis that computers can learn without being programmed to perform a specific task. Artificial intelligence researchers sought to know if computers could learn from data. Iterative machine learning is crucial because the model can adapt on its own as it is exposed to new data. Learning from previously seen data to deliver reliable, repeatable outcomes and results. The science is not new, rather it is gaining new momentum. Although many machine learning techniques have been in use for some time, it is still a recent invention to be able to automatically apply difficult mathematical calculations to large amounts of data [5].

The same variables that have increased the popularity of data mining and Bayesian analysis also explain the upsurge in interest in machine learning. increased availability of a wider range of data, less expensive and more powerful computing, more economical data storage, etc. All of this makes it possible to quickly and automatically create models that analyze more extensive and intricate data and deliver quicker, more accurate results, even at very large scales. Making an accurate model improves a company's chances of spotting lucrative opportunities and averting unidentified risks.

Machine learning technology is broadly divided into three main types: Supervise Learning, unsupervise learning, and reinforcement learning. In supervised learning, a set of data and corresponding results (label) is given to a machine learning algorithm to produce a machine-learning model. With this model, given a set of new data, the model predicts the results (label). Classification and regression are usually performed in supervised learning tasks. In unsupervise learning data without a label is provided to a machine learning algorithm. Based on the available data, the algorithm finds the pattern and groups them to provide labels. In contrast, reinforcement

Learning involves letting the machine learning model perform actions, and based on the outcome, the machine-learning model is either punished or rewarded [7].

Deep learning (also known as deep structured learning, deep machine learning hierarchical learning, or DL) is a subset of machine learning that relies on a set of algorithms to transform high-level abstractions in data using intricate structural model architectures and a multi-step non-linear modeling process. Deep learning is a member of a larger family of machine learning techniques built on the idea of learning how to represent data. Deep learning holds the potential of replacing manual feature creation with effective unsupervised or semi-supervised hierarchical feature extraction and learning algorithms.

Historically, Deep learning is an unsupervised learning algorithm. Its concept comes from the study of artificial neural networks. Deep learning is the development of artificial neural networks. Artificial neural networks imitate how nerve cells in the human brain function. It includes mathematically processing data using a sizable neural network (neurons have an activating function). The neurons are activated and their value is transmitted throughout the network when a specific threshold is reached based on the activation function. A multilayer perceptron with multiple hidden layers is a deep learning structure. By combining lower-level features to form higher-level and more abstract attributes or features, deep learning is used to discover distributed feature representations of data.

Deep learning is not like surface learning. Many learning methods are surface structure algorithms, which have certain limitations. For example, in the case of limited samples of complex features, the ability is limited. Its generalization to complex classification and clustering problems is affected by certain factors. Deep learning is learning through a deep nonlinear network structure to achieve the approximation of complex functions according to the representation of the distributed input data. Research in this area aims to make better representations and create models for learning these representations from large unlabeled data. In the case of sample sets, the essential characteristics of datasets are rarely studied [8].

Clustering is one of the unsupervise learning processes, which means that according to a certain criterion between samples, the result (the clustering, or the partition) is based solely on the object representation, the similarity measure, and the clustering algorithm under the unsupervised clustering process conditions. The clustering method employed can divide a large number of documents into clusters, that can be quickly understood by users so that users can grasp the content contained in a large number of documents faster, speed up analysis, and assist in decision-making. Extensive document clustering is part of the effective solutions to solve data understanding and information mining in massive texts.

The automatic clustering of textual documents (such as language corpora, text documents, tweets, emails, web pages, etc.) into groups based on the similarity of their content is known as text

clustering, or text document clustering. Descriptors from documents are extracted during the document clustering process. Word groups known as descriptors are used to characterize the contents of a cluster.

Historically, text clustering was developed to improve the performance of web browsers (such as Bing, Google, Baidu, Yandex, Yahoo, etc.) by pre-clustering the entire corpus to facilitate easy searching and retrieval of information. The general assumption is that documents that are similar to each other will tend to be related to the same query, so automatic cluster determination of such documents can improve retrieval by effectively expanding search requests. The main goal was to retrieve information and present it to the end-user. Document clustering was used to improve the ability to find related documents to a given query in its most simplified version [9].

Typically, documents belonging to a particular cluster are made to have high clustering quality by maximization of intra-cluster similarity and minimization of inter-cluster similarity. The bottom line is that feature vectors can be used to represent text documents as numbers. The index of similarity in text documents is compared by measuring the distance between these feature vectors. Objects near each other must belong to the same cluster while objects that are far from each other must belong to different groups [1, 10]. Generally, Text clustering can be used to achieve one of the following:

- Link similar text documents and remove duplicated documents.
- Create structure on the text data.
- Quickly get an idea about the content of a large collection of documents.
- Induce additional features (clusters) for the classification of text documents.

Topic modeling is a commonly used unsupervised learning text-mining technique for discovering semantic structures hidden in the body of text. The topic modeling algorithms work on the premise that since the theme of a document is on a particular topic, it is intuitively expected that a particular word will appear more or less frequently in the document. This intuition is captured by the topic model within a mathematical framework. This enables the study of a collection of documents and the identification of word groups and expressions that best describe the collection of documents. It also enables the identification of the topical balance of each document in relation to other themes. Topic models, also known as stochastic or probabilistic topic models, refer to statistical algorithms for discovering the potential semantic structure of large amounts of text [11].

Therefore, the research topic of this thesis, text clustering and topic modeling of tweets about the COVID-19 vaccination using machine learning, deep learning, and NLP – (natural language processing), may be carried out by applying various deep learning and machine learning techniques.

The Research Problem

Since the World Health Organization classified the COVID-19 outbreak as a pandemic, several efforts have been made by different disciples to contribute to combating the deadly pandemic. Artificial intelligence and data science are some of the new technologies used by different disciplines to fight coronavirus disease. Some of the roles played by AI in the healthcare discipline include early detection, monitoring, and prediction of the infection, contact tracing and monitoring the treatment of the infection, development of drugs and vaccines, etc. In the robotics field, drones help in delivering critical medical supplies and hence prevent the spread of the virus. In addition, Robots are also used to sterilize, supply and distribute food, and perform other tasks. A.I. and data science-powered algorithms like machine learning and natural language processing can help predict impending epidemics by analyzing information from a multitude of sources and tracking over a hundred infectious diseases. In the future, A.I. could be even used for predicting human behavior and potential outbreaks through social media data [12, 13].

A detailed examination of the data on coronavirus data for data science, from a variety of sources, including our world in data, the World Health Organization, data world, world meters, and Reuters, reveals that the number of data available is in the hundreds of millions. Without labels, text and image data are highly unstructured. Data containing various entries that cannot be categorized or fitted to a straightforward model are referred to as unstructured data. Unstructured data, in a nutshell, is information that is not organized or without an established data model. Although unstructured data frequently contains a lot of text, it can also include facts, dates, and figures. To generate insight from such data, the available data needs to be converted to a structured format. Structured data are data that are highly organized. In other words, structured data can best be described as data that conforms to a data model, has a well-defined structure, follows a consistent sequence, and is easily accessible and used by human or computer programs. Structured data is usually stored in a well-defined schema, such as a relational database.

Twitter is a popular online platform for microblogging that disseminates over 400 million messages daily and offers a wealth of information on virtually every sector, including business, sports, entertainment, and health. The main attraction is how easily accessible it is. Twitter is simple to use for information sharing and gathering. Twitter gives politicians, celebrities, huge corporations, non-governmental organizations (NGOs), etc. unprecedented access to millions of individuals delivering the most recent news. Twitter is also an important data source for large corporate business models. All of the above properties make Twitter an ideal place to collect and analyze real-time updates and investigate real-world situations.

Research by Kwak, H., Lee, C., Park, H., and Moon, S. has tried to answer the question of what is Twitter. They researched Twitter's topological properties and its effectiveness as a cutting-edge information-sharing platform. A non-power law follower distribution with a small effective

diameter and little reciprocity was discovered after the topology study. All of them deviate from the established characteristics of human social networks. They ranked Twitter users by follower numbers and PageRank to determine who was influential, and they discovered two similar rankings. The retweet ranking differs from the first two rankings. This demonstrates the disconnect between the number of followers and the influence of a user's tweets' popularity. They examined tweets about the trend's top subjects and published reports on user participation and behavior over time. The majority of the topics (85% or more) are headline news or are essentially permanent news, as shown by the classification of trending topics by active period and tweets [14].

By observing the microblogging phenomenon and analyzing the topological and geographic characteristics of Twitter's social network, Java, Akshay, Xiaodan Song, Tim Finin, and Belle Tseng attempted to provide an answer to the question "why do we twitter." They discovered that individuals use microblogging to discuss their daily activities and to look for or distribute information. Additionally, the study examined user intents related to communities, and the findings revealed that users connect when their intentions are comparable [15].

To better understand how people connect with one another in the digital age, many researchers have employed Twitter data. In order to distinguish between elite users, such as celebrities, bloggers, and representatives of media outlets and other formal organizations, and regular users, Wu, S., Hofman, J.M., Mason, W.A., and Watts, D.J. investigated some long-term questions in media communication research concerning information generation, flow, and consumption. According to the survey, celebrities are the most followed users on Twitter, the media provides the most content, and only 20,000 elite users produce around 50% of the URLs that are consumed. The study also discovered that there is high homophily within categories at the contact level, with celebrities listening to celebrities and bloggers listening to bloggers, etc. However, bloggers broadcast more information than the other groups as a whole [16].

Paul, M. and Dredze, M., analyzed user messages on social media to measure various demographic characteristics, including public health interventions for more than 12 illnesses, including allergies, obesity, and insomnia. The study compared qualitative evaluations of model output with quantitative correlations with public health statistics. This finding indicates that Twitter can be widely applied to public health research [17].

Researchers have also utilized Twitter to analyze human behavior during the COVID-19 outbreak. Using Twitter data, Simranpreet Kaur and Pallavi Kaul provided a paradigm for analyzing the behavioral shifts and emotional dynamics among Twitter users throughout the pandemic. The system processes the tweets using natural language processing and social network analysis approaches to get sentiments. The study's findings demonstrated a strong relationship between Twitter users' emotional traits, infection, and fatality rates [18]. About 20 million tweets were analyzed by Lwin, May Oo, Jiahui Lu, Anita Sheldenkar, Peter Johannes Schulz, Wonsun

Shin, Raj Gupta, and Yinping Yang to evaluate global trends in fear, rage, grief, and joy as well as the narratives that underlie these feelings during the COVID-19 epidemic. The study found that throughout the pandemic, people's feelings dramatically fluctuated between fear and rage, with melancholy and happiness also showing up. Word clouds' findings indicate that concerns about COVID-19 test shortages and medical supply shortages spread like wildfire. At the start of the pandemic, xenophobia predominated; now, the debate centers on the stay-at-home warnings. Subjects associated with joy included expressions of thankfulness and excellent health, while topics related to sadness were emphasized by the loss of friends and family members [19].

Twitter can also be used to track public health, including tracking, forecasting, and responding to events. The authors of the study by AbdAlrazaq, A., Alhuwail, D., Househ, M., Hamdi, and Shah, Z., identified the top tweets related to the COVID-19 pandemic. Out of the roughly 2.8 million tweets examined, the study discovered that 167,073 distinct tweets from 160,829 distinct people satisfy the selection criterion. Twelve themes were found in the analysis, which was divided into four main themes. What causes it, how it affects people, nations, and economies, and how can it be reduced? [20]. According to a study by Pershad, Y., Hangge, P.T., Albadawi, and Oklu, Twitter is currently the most widely used social media platform for healthcare communication. The authors took into account the potential for using Twitter to exchange research information to advance treatment and care, in particular [21]. In a survey of the literature on the topic, Jordan, S.E., Hobet, S.E., Fong, I.C.H., Liang, H., Fu, KW, and Tse, Z.T.H. discussed the use of Twitter or other short text datasets for data mining in public health applications. Each approach to leveraging Twitter data for public health monitoring was examined in this study. To better monitor public health, papers that sorted Twitter content by user or tweet were chosen for review. Only texts released between 2010 and 2017 were taken into account. The categorization of Twitter content utilized in the published publications under evaluation varies. This cutting-edge review illustrates the enormous potential of employing Twitter for maintaining public health [22].

Recently, there has been an international campaign to immunize everyone against the unique COVID-19 pandemic to stop its spread, lessen the risk of coronavirus disease variations, and create herd immunity. The Our World in Data website reports that 61.4% of the world's population has gotten at least one dose of the COVID-19 vaccine. Globally, 10.22 billion doses have been given out, and 23.37 million doses are given out per day (listing as of Feb. 7, 2022). The amount of tweets regarding the COVID-19 vaccine that is currently available exceeds hundreds of millions, according to the various data sources for COVID-19 vaccine data. Because of the timing constraints, particularly in terms of delivering results as soon as feasible, classifying the data becomes challenging. However, for efficient decision-making, it is necessary to comprehend the substance of this data [4].

It is crucial to first group the tweets on the COVID-19 vaccine into clusters to rapidly comprehend their contents. Next, use topic modeling to quickly comprehend the many themes contained in each cluster. The research question is thus as follows: given a large volume of coronavirus vaccine tweets, how do you cluster these tweets based on similarity and as well as perform topic modeling of the tweets to understand the contents of the tweet as soon as possible knowing that there is a time constraint and the volume of the tweet is enormous?

Research Aims, Objectives, and Hypothesis

At the document, phrase, or word level, text clustering is possible. Documents on the same subject are grouped using the document-level field. Search engines, emails, news stories, and other types of documents can all use text document clustering. Sentences from various documents are grouped using sentence-level. An example of sentence-level grouping is the analysis of tweets. Groups of words, or phrases with a common topic, make up the word level. Putting together a word-level cluster is made simple by gathering synonyms for particular words. For instance, the WordNet dictionary organizes English words into sets of synonyms known as synsets.

There hasn't been a lot of research done on text clustering in coronavirus vaccination tweets. This is partially caused by the sparseness and high dimensionality of the COVID-19 vaccine tweets data.. Lyu, Joanne Chen, Eileen Le Han, and Garving K. Luli did a study closely related to this discourse. In order to understand the salient changes in topics and sentiments over time and the public perceptions, concerns, and emotions that may affect the achievement of herd immunity goals, the researchers studied topic modeling and sentiment analysis in the public COVID-19 vaccine-related discussion on social media using Twitter. Analysis of the data revealed that despite swings in sentiment, weekly mean sentiment scores were generally growing in positivity. A further study of the emotions revealed that trust predominated, followed by anticipation, fear, sadness, etc. On November 9, 2020, Pfizer claimed that its vaccine is 90% effective, and this is when the trust feeling peaked [23].

This thesis uses unsupervise learning approaches, machine learning, natural language processing, and deep learning to determine the number of clusters and the topics inside the clusters that are prevalent in the coronavirus vaccination tweets. By comparing the resulting clusters and topics to identify the best clusters and most representative subjects of the Covid-19 vaccine tweets using deep learning and machine learning algorithms, the thesis also intends to justify the need for deep learning-based methods for clustering and topic modeling. The research objectives are listed below:

1. To extract meaningful information and insights from COVID-19 vaccine tweets by doing exploratory data analysis of the tweets.

2. To look into the effectiveness of deep learning and machine learning models for grouping educational tweets from vaccine data.
3. The thesis also shows how word embeddings can be employed to improve the precision of these models.
4. To identify the prevalent topics within the COVID-19 vaccine tweets using topic modeling algorithms.
5. The thesis makes use of multiple deep learning and machine learning-based evaluation criteria to assess the performance of the models.
6. The thesis shows how data can be summarized using the output of the constructed model and its new labels for supervised machine learning classification and pattern recognition purposes.

The research Hypotheses are listed below

1. Text clustering and topic modeling on coronavirus vaccine tweets are more likely to reveal trends in Covid-19 vaccination efforts.
2. Text clustering and topic modeling on coronavirus vaccine tweets are more likely to reveal the major news that dominated the tweets.
3. Text clustering and topic modeling on coronavirus vaccine tweets are more likely to reveal the types of vaccines that are available for vaccination within a locality.
4. Text clustering and topic modeling on coronavirus vaccine tweets are more likely to reveal structure and similarity among the created COVID-19 vaccine tweet clusters.
5. Text clustering and topic modeling on COVID-19 vaccine tweets are more likely to reveal trends in the progress of the vaccination exercise.

The Significance of the Study

The COVID-19 pandemic has affected many virtually every sphere of our lives from science and technology to how we interact with one another as well as institutions around the world, resulting in reduced productivity in many areas and programs. However, the influence of the pandemic has opened several new research funding lines for government agencies around the world. The COVID-19 pandemic has developed a new and improved form of scientific communication. Examples are the amount of data published on the pre-print servers and its validation on social media platforms before it was officially peer-reviewed. Scientists quickly review, edit, analyze, and publish manuscripts and data. This intense communication may have enabled an extraordinary level of collaboration and efficiency between scientists.

COVID-19 vaccination provides a way out of this stage of the pandemic. Many scientists believe that without it, natural herd immunity would not have been sufficient to bring society back to normal, and this would have resulted in extreme death. This has been confirmed by many health

agencies, including WHO. In scenarios where vaccines are not available, strict codes of conduct must be maintained for the foreseeable future. Therefore, there is a need for more research on COVID-19 tweets to understand how it affects people and monitor the progress of vaccination exercises around the world. Social media platforms like Twitter are one of the ways to get feedback on vaccine hesitancy, acceptance, side effects, and progress.

This research will be beneficial to health care professionals, media houses, academia, computer scientists, etc. Exploratory data analysis of the coronavirus vaccine tweets will provide new information to researchers and other scholars on the types of clusters that can be formed from the coronavirus vaccine tweets and the content of those clusters. Social science researchers will benefit from this study by looking at and understanding social perception and response to the vaccine. This research will also help computer scientists to understand various unsupervised deep learning and machine learning methods and techniques that can be applied to COVID-19 vaccine tweets and their efficacy. This novel research will also create new insight into the use of unsupervised deep learning models (deep clustering) for COVID-19 research. This research will help medical personnel to get feedback, understand the general themes in the tweets, and find out ways to improve vaccination to achieve herd immunity. Medical practitioners such as doctors and pharmacists can better understand the complaints about the various vaccines and ways to alleviate them.

Additionally, by demonstrating how to analyze data on tweets related to the COVID-19 vaccine, the analysis and methodology given in this study will add to the body of knowledge and serve as a standard for managing pandemic tweets in the future. Data insights will be helpful to evaluate, learn from, and prepare for future management and distribution of vaccinations during a pandemic. Gained knowledge can also be applied to decision-making for pandemic management in the future.

The Scope of Study

With the increase in the amount of social media users year on year and its use in everyday communication at both individual level and organizational levels, there has been a corresponding increase in its incorporation to understand what individuals and organizations are saying as regards the spread of the coronavirus disease, especially during the COVID-19 pandemic. Recently there have been major efforts both globally and nationally to produce vaccines that can curb the spread of the pandemic and as well return life to normalcy. Therefore, this study will focus on the use of the Twitter microblogging site to understand what individuals and organizations are saying as regards the new vaccination drive to end the pandemic.

Whilst sentiment analysis, emotions analysis, and classification of various COVID-19 data have been well documented in the literature, clustering and topic modeling of coronavirus vaccine

data have not been well understood. This study aims to report on the clustering and topic modeling of coronavirus vaccine tweets using various machine learning and deep learning algorithms as well as natural language processing techniques.

The scope of the study is limited to COVID-19 vaccine tweets collected from Twitter within a fourteen months cycle from December 2020 when the first COVID-19 vaccine was announced to February 2022 one year and two months after the vaccine was announced using the COVID vaccine hashtag. There are various algorithms used for text document clustering but the problem of short text document clustering is a hard one due to its sparsity and high dimensionality. Therefore, the deep learning and machine learning algorithms considered therein are those that have specifically worked well with short texts in literature. Since the data have no labels, the evaluation algorithms used are those that do not require ground truth labels.

Further, the study also involves an analysis of the clusters to understand the topics and context within the clusters. The time scope for the text clustering models also applies to the topic models. Topic modeling of short texts like Twitter tweets also shares the same characteristics as text clustering with the sparsity of tweets and high dimensionality a recurring theme. Therefore, the topic modeling techniques used in this study are those that have worked well with short texts in literature.

The structure of this research thesis is as follows: The existing literature and related work are covered in Chapter 2, the research methodology and methods used for topic modeling and cluster analysis are covered in Chapter 3, and the research experimental results and findings are covered in Chapter 4, and the analysis and interpretation of the research results and findings are covered in Chapter 5. The conclusions of this research thesis are presented in chapter 6, along with suggestions and recommendations for additional studies.

2. LITERATURE REVIEW

The chapter presents a detailed review of the literature on clustering or cluster analysis, text clustering, dimensionality reduction, and topic modeling. The chapter also presents relevant literature on applications, types, and use cases of clustering, text clustering, and topic modeling. The chapter also highlighted the existing gap in COVID-19 literature and justified the need to incorporate text clustering and topic modeling into COVID-19 literature.

2.1. Clustering

The unsupervised categorization of patterns (data points, observations, feature vectors, etc.) into groups is known as clustering. Clustering has been addressed in a variety of circumstances and by scholars from a variety of fields. This indicates its broad popularity and utility as a phase in exploratory data analysis. In recent years, clustering has become an increasingly prominent topic. This is because of the influx of data from numerous fields. However, due to a lack of communication across these communities, similar theories or algorithms were frequently recreated, squandering time and money.

Clustering or cluster analysis was a major early educational focus for Edwin Diday. Principles of Numerical Taxonomy, a 1963 monograph by Sokal and Sneath, stimulated international research and interest in clustering techniques and sparked the publication of books such as "Les bases de la Klassifikation Automatique" by Lerman, “Application Cluster Analysis” by Anderberg, "Mathematical Taxonomy" by Jardine and Sibson, “Automatic Classification” by Bock, "Application Cluster Analysis," by Anderberg, "Cluster Analysis" by Bijnen, “Classification Empirical Methods” by Sodeur, “Cluster Analyse Algorithmen” by Spath, “Numerical Classification Problems and Methods” by Vogel, and “Clustering Algorithms” by Hartigan all in the 1970s. As a result, the fundamental issues and techniques of clustering are well known to the larger scientific community in statistics, data analysis, and its applications [24].

A group of objects known as cluster analysis or clustering can be utilized in a number of practical applications, including data/text mining, voice mining, image processing, web mining, etc. In his article, Bijuraj LV. outlined the significance of clustering in the actual world as well as the various uses for the technology [25]. In a different study, Lee, R.C., examined a variety of clustering analysis approaches and their applications [26].

Because clustering is a challenging combinatorial problem, it takes a long time for general notions and techniques to spread throughout diverse populations with diverse presumptions and circumstances. Additionally, practitioners and those with little to no experience with cluster analysis find the various phrases particularly perplexing. In this area, precise and thorough

information is needed. A research article on clustering was written in 1999 by Jain, A.K., Murty, M.N., and Flynn, P.J., to standardize definitions among practitioners in the community. The white paper provided helpful guidance and references to fundamental ideas that are accessible to a large group of cluster practitioners as well as an introduction to pattern-clustering approaches in terms of statistical pattern recognition. The study also provided a classification of clustering methodologies to pinpoint prevalent problems and current developments. Additionally, it discussed some significant uses of clustering techniques in the fields of object recognition, image segmentation, and information retrieval [27].

In order to address this need, among other things, Xu, Rui, and Don Wunsch wrote a book on clustering in 2008. The book aims to provide a thorough and systematic description of important clustering algorithms that are rooted in computer science, statistics, machine learning, and computational intelligence, with a focus on current breakthroughs in recent years. The 11 chapters in the book cover a range of topics, including the fundamentals of similarity metric, cluster analysis, and cluster evaluation, as well as a number of clustering algorithms, including partitioned clustering, hierarchical clustering, kernel-based clustering, neural network-based clustering, large-scale data clustering, sequential data clustering, and high dimensional data clustering. Additionally, it offers a thorough reference as well as examples of recent applications, including web document clustering and bioinformatics [28].

Von Luxburg, Ulrike, Isabelle Guyon, and Robert C. Williamson, studied whether clustering was an art or science, but concluded that viewing clustering as a domain-independent method is nonsensical. They used a general science-based method that does not rely on a particular clustering algorithm to see if the qualities of different clustering algorithms can be compared. The study argued that the main obstacle is the difficulty of evaluating clustering algorithms without considering the context. The study tried to answer the question “Why do users cluster their data first, and what does the user use the clustering for afterward?” and as well argued that clustering should not be treated as an application-independent math problem but should always be considered in the context of its end-use. The research suggested that different methods for evaluating clustering algorithms needed to be developed for different clustering applications. The study emphasized that for researchers to have a real impact on their applications. Then it is time to take a step back from a purely mathematical and algorithmic point of view because what is missing is not a "better" clustering algorithm, but a problem-centric perspective to make meaningful cluster evaluation possible [29].

In a survey work on data clustering, Jain AK. gave a quick introduction to the topic, listed all of the existing clustering techniques, and highlighted the most significant ones. The study examined difficulties and fundamental issues in creating clustering algorithms, and it also identified

new and fruitful research areas such as ensemble clustering, semi-supervised clustering, large-scale data clustering, and simultaneous feature selection during data clustering [30].

Gan, G., Ma, C., and Wu, J., wrote a book named data clustering: Theory, Algorithms, and Applications. The book described over 50 algorithms for clustering data and groups according to the underlying methodologies such as search-based, center-based, graph-based, grid-based, model-based, and density-based. It also describes hierarchical clustering and fuzzy clustering. The book provided pseudocode for each algorithm with the relevant mathematical concepts briefly presented, and the theorems generally presented, but not proven. Although references to related publications allow for exploration of more specific algorithms and concepts. Part I of the book is generally about data types, data standardization, similarity measurement, and visualization (more accurately, algorithms for visualization of high-dimensional data such as t-SNE and MDS). Part II contains a list of various clustering algorithms and a chapter on clustering evaluation. Page 277 upwards contains various indexes such as Rand and Dunn indexes. Finally, Part III introduces software for clustering, and Part IV focuses on two applications: gene expression data clustering and variable annuity clustering guidelines. Chapter 19 shows the packages available in R and Python. However, the latest packages can easily be found online [31].

Numerous studies have been conducted on the improvement of clustering algorithms over the past 40 years. As a result, numerous strategies that are based on various methodologies have been suggested in the literature. Clustering techniques were described by Milligan, G.W., and Cooper, M.C., who also concentrated on the algorithm's performance and the ensuing effects on applied research. After reviewing the clustering literature, the researchers described the clustering process in a 7-step framework. The study identified four main types of clustering methods, which are characterized by the representative algorithms as hierarchical, partitioning, overlapping, and ordination. Validating such algorithms is related to the problem of determining the function of the method that restores the cluster configuration known to be present in the data. The validation approaches employed in the research include mathematical derivations, empirical dataset analysis, and Monte Carlo simulation methods. The study went on to explain cluster analysis interpretation and inference techniques. Testing significant cluster structures and the issue of counting the number of clusters in the data are examples of inference approaches [32]. Omran, G.M., Engelbrecht, P.A., and Salman A., took another approach in their review paper on clustering methods. The clustering methods are divided based on their degree of membership to a particular cluster. The various typical clustering approaches are fuzzy clustering or soft clustering, hard clustering or partition-based clustering, Gaussian expectation maximization, and Hybrid approaches. In addition, the study also examined different cluster validation indices as well as approaches for automatically determining the number of clusters. Additionally, various heuristic solutions to the clustering problem were researched [33].

One of the oldest studies on clustering algorithms is Johnson, SC's study on the hierarchical clustering approach. He studied techniques for dividing objects into an optimally homogeneous group based on an empirical measure of similarity between these objects, which was attracting attention in various fields at the time. The paper developed useful correspondence between such hierarchical system clusters using specific types of distance measures. The method used the matrix element twice, by finding the smallest matrix element that is not equal to zero and then forming the maximum or minimum matrix elements. Both processes are executed without the knowledge of the underlying data except the rank order. The research result produced two computationally fast clustering techniques that are Immutable under monotonous conversion of data. One of the methods (minimum) in its defined sense form optimally "connected" clusters while the other (maximum) forms optimally "compact" clusters [34]. Murtagh, F. surveyed recent advances in hierarchical clustering and presented a well-detailed progress report. The study described a general framework for hierarchical agglomerative clustering. The research disproved the assertion that hierarchical clustering often requires pairwise stitching between objects, so the complexity of these clustering techniques is often at least O (N^2) by integrating an efficient nearest neighbor searching algorithm into the clustering method, which offered many improvements over the algorithms that are currently in widespread use. The progress report describes new algorithmic approaches in this area and outlines the latest results [35]. Murtagh, F. and Contreras, P., presented an overview of various hierarchical clustering algorithms in their research work. They looked at aggregated hierarchical clustering techniques and described other effective software implementations and R. A hierarchical self-organizing map and a mixed model were also taken into account in the investigation. The study examined hierarchical density-based clustering while reviewing grid-based clustering. The discussion ended with a discussion of a recently created, extremely effective (linear) hierarchical clustering algorithm, which is also known as a hierarchical grid-based algorithm [36, 37].

Partitioning Data clustering also referred to as partitioning-relocation clustering, is an important technique. In the literature, a number of partition clustering approaches and strategies have been established. In his article, Berkhin P reviewed both different data mining and partition-clustering strategies. He proposed that a crucial clustering strategy that many clustering algorithms are built on is the partition-clustering approach. A dataset is immediately partitioned into a number of comparatively unrelated clusters using partitioned clustering. An integer number of clusters that optimize a specific reference function is a result. The optimization of the reference function is an iterative relocation process, and it can be used to emphasize the local or global structure of the data. Since the clusters in this method of clustering are thought to be relatively prime and do not have support for intersections [38].

Chapter two of Kutbay, U.'s book on recent applications of data clustering throws more light on the partition or partition-based clustering. The chapter described some common techniques used

in partitioned clustering. The chapter further explained that partition clustering uses an iterative process to assign a collection of data points to the K cluster. Predefined reference function (J) assigns a datum to the kth number set. Clustering can be performed by setting this reference function value to k sets (maximization and minimization calculations). The definition of the clustering process's criterion function in terms of mean squared error, the sum of squared errors, etc. was covered in the chapter. The conclusion that partition clustering is an iterative process was aided by this. The evolutionary process is one of the search methods used in partition clustering that enables one to find the global or local optimum by parameter optimization. In order to tackle the optimization problem, search approaches add additional parameters than the standard partition clustering process. Therefore, there is no theoretical basis for selecting the ideal course of action. On various types of data, any of the suggested strategies can produce the best results [39].

Density-based clustering, which is utilized for cluster analysis in a variety of circumstances, was defined by Kriegel, P.H., Kröger, Sander, and Zimek in their review work. According to the article, density-based clustering identifies a cluster as a high-density area that is separated from a low-density area so that it can take on any shape that is required in the data space. Different groups have developed a number of formalizations that are similar to density-based clusters. These methods are typically predicated on the fundamental notion that a cluster is a connected, high-density component. The definitions of density and connectedness are typically the only areas where the various methodologies diverge. In the study, multiple techniques for computing density-based clusters in various settings and communities were highlighted. The study also revealed that the majority of the early methods for cluster statistics did not emphasize efficiency problems. Following that, several methods for massive databases were created in the database and data mining sectors. The research article also suggested certain modifications and additions to the fundamental idea of density based clustering [40].

The most crucial ideas and algorithms for density-based clustering were summarized by Martin Ester in the book density-based clustering. The most crucial elements of density estimates, connection definitions, and data structures for effective implementation were covered in the book. Because it makes no assumptions regarding the quantity or distribution of clusters, density-based clustering can be regarded as a nonparametric technique. The data space is divided into sparse and dense regions, which are joined by dense-based clusters. The book went on to explain that density-based clustering methods calculate the density by counting the number of points within a given radius and treating any two points that are close to one another as related. The book's later chapters examined sophisticated density-based approaches applied in subspace clustering, data streams, uncertain data, and network data clustering [41].

A review study by Campello, R.J., Kröger, P., Sander, and A. Zimek was updated to reflect recent developments in the subject. Density based clusters are separated from one another by

continuous regions with low object density, according to the researchers' refined definition of density based clustering, which states that a cluster is a collection of data objects distributed in a contiguous area of data space with high object density. This new definition implies that the data objects located in the sparse area are typically regarded as noise or outliers. The review article discussed hierarchical density-based clustering approaches, semi-supervised clustering methods, and traditional algorithms for deriving flat divisions of density-based clusters. The review study discussed various unresolved difficulties in its conclusion. The review paper concluded by discussing several unresolved problems in relation to density based clustering [42].

In his study, Mark Girolami, demonstrated how to unsupervisedly divide a sample of data and how to calculate the number of distinct clusters that produce the data. The idea that performing a non-linear data transformation into a high-dimensional feature space increases the probability of linear separability of the patterns in the transformed space and simplifies the associated data structure was used to explore the idea of data clustering in the kernel defined feature space. This is connected to the principal component analysis of features performed unsupervised and the support vector classification approach employed by the Mercer kernel representation of dot products in feature space. The article also demonstrated how to estimate the number of clusters that are specific to the data using the eigenvectors of the kernel matrix that define the implicit mapping, presenting a computationally straightforward iterative strategy for successive feature-spatial partitions of the data [43].

According to F. Camastra and A. Verri, the kernel technique is an algorithm that, by substituting a suitable positive-definite function for the inner product, accomplishes an implicit non-linear mapping of input data to a high-dimensional feature space. The research presents a clustering kernel approach that was modeled after the traditional k-means methodology. In this manner, a single class of support vector machines is used repeatedly to enhance each cluster. The proposed method outperforms well-known clustering algorithms like neural gas, k-means, and self organizing maps – (SOM) on a synthetic data set and three UCI benchmarks, the Wisconsin breast cancer dataset, the iris dataset, and the Spam dataset. Its empirical convergence has also been consistently verified [44].

A survey of spectral and kernel clustering techniques was presented by Filippone, Camastra, Masulli, and Rovetta. An overview of the kernel and spectral clustering methods, two strategies that can produce nonlinearly separated hypersurfaces between clusters, was the aim of this paper. The kernel versions of numerous well-known clustering methods, including neural gas, k-means, and self organizing maps, are used in the described kernel clustering methodologies (SOM). A brief history of kernel approaches was also emphasized in the survey piece. According to the publication, Aizerman et al. first used kernel functions in machine learning in 1964. Support Vector Machines (SVMs), which perform better than traditional classification methods for particular applications,

were first introduced by Cortes and Vapnik in 1995. The popularity of SVMs led to the expansion of kernel use in other learning techniques (such as kernel-PCA) [45].

A new clustering method based on the kernel method was proposed by Piciarelli, Micheloni, and Foresti in their research article. In contrast to many popular clustering techniques, the suggested method has the benefit that, unlike the partitioning or expectation-maximization approaches, the number of clusters is automatically recognized without the requirement for initial estimation. The technique uses normalized kernel space geometry to automatically determine the right number of clusters. By doing so, the initial estimation of this parameter that many common algorithms demand can be avoided [46].

Hans-Hermann Bock, in his research article, described how neural network (NN) approaches such as Hopfield Networks, Multilayer Perceptron, and Kohonen's Self-Organizing Maps (SOMs) could solve clustering problems. The paper further emphasized the close relationship between the NN approach and traditional clustering techniques. In particular, Anouar et al, 1997. Showed how SOM is derived by stochastic approximation from a new continuous version (k-criterion) of the finite sample clustering criteria. In this context, the research determined the asymptotic behavior of the Kohonen method, and as well designed a new finite sample version of a k-means type SOM method, and proposed a generalization that varies along with the classical “regression clustering”, “principal component clustering”, and “maximum likelihood clustering” [47].

Du, K.L., attempted to distinguish between clustering methods based on competitive learning or statistical model identification - (McLachlan & Basford, 1988). The paper offered a thorough analysis of competitive learning-based clustering methods using a neural network strategy. Additionally, the study described a number of competitive learning-based clustering neural networks, including learning vector quantization - (LVQ), self-organizing maps - (SOMs), neural gas, and ART models, as well as variations of these clustering techniques, including c-means, fuzzy c-means (FCM), and mountains/subtractive clustering. Additionally, it discussed auxiliary subjects as underused problems, robust clustering, fuzzy clustering, hierarchical clustering, clustering based on non-Euclidean distance measurements, supervise clustering, and cluster validity. The study included two examples to demonstrate the use of the clustering approaches [48].

F. Roohi defended the need for a clustering method based on neural networks. He underlined that attempts to replicate the learning and function of the human brain have replaced the development of powerful computers and non-learning programmed closed systems in the fields of artificial intelligence and computing. The development of a model with an integrated self-learning mechanism that can continuously learn from the data and adjust the rules and outputs in accordance is required to do this. According to the research, numerous attempts have been made to achieve this goal, but neural networks have gained universal acceptance because of their durability and capacity to simulate the human brain. Numerous artificial neurons are combined and connected to form these

networks, which imitate the processes and capacities of human learning. They are the cornerstone of numerous clustering programs and the foundation of artificial intelligence. According to the article, neural networks are frequently employed for the majority of data analysis, decision-making, design, and prediction tasks, which are the foundation of clustering issues. Artificial neural networks (ANN) are commonly employed today in both supervised and unsupervised learning and are highly helpful for clustering tasks, especially if the number of clusters is not known in advance, according to the research. The paper attempted to explain ANN, its elements, and how it is used to carry out various clustering tasks. The ANN clustering techniques discussed include Adaptive Resonance Theory (ART), self-organizing feature maps (SOFM), and learning vector quantization (LVQ) [49].

One of the most well-liked and current clustering methods in recent years is spectral clustering. It is frequently more effective than conventional clustering methods like partition-based algorithms and is simple to apply. It can also be solved efficiently using typical linear algebra tools. Donath and Hoffman (1973), who initially suggested creating graph divisions based on the eigenvectors of an adjacency matrix, are credited with the invention of spectral clustering. In the same year, Fiedler (1973) advocated utilizing the second eigenvector of the Laplacian graph to split the graph after finding that the two divisions were closely related to it. Since then, numerous times in various communities, spectral clustering has been discovered, rediscovered, and expanded. Such studies include those by Bolla (1991), Phothen et al. (1990), Hagen and Kahng (1992), Simon (1991), Van Driessche and Roose (1995), Hendrickson and Leland (1995), Barnard et al. (1995), Guattery and Miller (1998), and Spielman and Teng (1996). In 1996, Spielman and Teng provided a history of spectral clustering. Shi and Malik's work helped spectral clustering become well-known in the machine learning community (2000).

One of the intriguing alternatives to the spectral approach of clustering that has surfaced in several fields was examined by Jordan, M.I., Ng, Y.A., and Weiss, Y. The upper eigenvectors of the matrix, which are formed from the separation between the points, were used by the researchers in this case as characteristics for clustering. The researchers aimed to address spectral clustering's persistent open concerns despite the numerous empirical achievements of spectral clustering approaches and algorithms documented in the literature. Eigenvectors are used in a variety of algorithms in significantly different ways. Second, a lot of these techniques need proof that they provide accurate clustering. The article proposed a straightforward spectral clustering approach that can be used with MATLAB and implemented in just a few lines of code. The method specified the conditions under which the algorithm is supposed to operate and analyzed the algorithm using methods from matrix perturbation theory. For several challenging clustering issues, the spectral clustering technique also produced very positive experimental results [50].

Bach, F. & M. Jordan Based on the measurement of the discrepancy between a given partition and the spectral relaxation solution of the least normalized intersection problem, the authors of this study developed a novel cost function for spectral clustering. According to the paper, "spectral clustering" refers to a class of methods that partition points into spatially distinct clusters using the Eigen structure of the similarity matrix. While points in various clusters are not very similar to one another, points within the same cluster are quite similar. The study also found that minimizing in terms of the similarity matrix results in an algorithm for learning the similarity matrix, while minimization in terms of the cost function associated with partitions results in a new spectral clustering technique. In addition, the power method-based cost function approximation was created for computing the eigenvectors [51].

A research paper lesson on spectral clustering was written by U. Von Luxburg. It may seem a little bizarre and unclear what spectral clustering truly achieves or why it works in the first place, but the study aimed to demystify it by offering intuition about it. The paper introduced the most popular spectral clustering algorithms and described several Laplace graphs as well as their fundamental properties. It also employed various methods to create these algorithms from scratch. Additionally, the study discussed the benefits and drawbacks of several spectral clustering approaches [52].

Liu, J. & J. Han They introduced several definitions of formulations of the graph Laplace operator and the precise processes for various types of spectral clustering approaches in their research, which was included in the book chapter on Data Clustering algorithms and applications. Both the spectra of unnormalized and normalized Laplacian matrices were examined in the study. The ability to analyze the spectrum of the Laplacian graph to provide clustering results is referred to as a spectrum. The Laplace Eigen map approach, a highly effective dimensionality reduction technique that is closely related to the generation of spectral clusters based on spectral graph theory, was also investigated in this study. Principal component analysis, one of the most widely used dimensionality reduction approaches, was also investigated. By transforming the initial high-dimensional data into a low-dimensional linear subspace that encompasses the main eigenvectors of the data's covariance matrix, the principal component analysis offers a solution. Additionally, the essay provided a broad overview of a standard spectral clustering technique, which calls for the construction of an adjacency matrix and the Eigen decomposition of the related Laplacian matrix [53].

In this article, Ruspini, E.H. proposed a general mathematical formulation of the data reduction and clustering process that allows for a unified view of data reduction techniques. The procedure is considered a mapping or conversion of the original space to a "representation" or "code" space, but with some limitations. As part of this formulation, the current clustering techniques and three other techniques for data reduction are discussed. To avoid some of the

problems of current clustering techniques and to better understand the structure in the original data, new methods of representing reduced data based on the "fuzzy sets" concept were proposed. The proposed approach is essentially free of shape and size issues in relation to other clustering techniques and, with proper constraints, provides better information about the structure of the dataset. In addition, formulating secondary and optimal conditions is easier in this context than the usual set description [54].

Ruspini, E.H. in another study presented several ways to find fuzzy clusters. Most of the work dealt with numerical techniques that help solve fuzzy clustering problems. The first two expressions described in the research do not provide an acceptable fuzzy classification but provide a good starting point for minimizing the third function. The last method gives a good fuzzy classification, but it requires a good initial approximation to converge to a better classification than the previous process. The third function suggested changes to keep partitions in a set of two or more. The article presented numerous computer experiments and methods under consideration that can achieve faster convergence [55].

A critique of the fuzzy set theory applied in cluster analysis was written by M.S. Yang. The research delved deep into the background of fuzzy clustering. The study concluded that although Zadeh's fuzzy set theory from 1965 supplied a general understanding of membership uncertainty as given by the membership function, it was wrong when it came to class membership. The study also noted that Bellman, Caraba, Zade, and Ruspini's studies from 1966 and 1969, which opened the door to the study of fuzzy clustering, were the first to propose using fuzzy set theory in cluster analysis. Fuzzy clustering is one of the most crucial cluster analysis approaches, and the study also examined algorithms that have been extensively investigated and utilized in a number of significant domains. The article gave a summary of the three types of fuzzy clustering. Fuzzy clustering, which is based on fuzzy relationships, is the first category. Fuzzy clustering based on the objective function is the second. Finally, a description of the nonparametric classifier, the fuzzy generalized nearest neighbor approach, is provided [56].

De Oliveira JV, and Pedrycz W., in their book advances in fuzzy clustering and its applications, defined fuzzy clustering as a mature and dynamic field of study with highly innovative and advanced applications. The book summarizes fuzzy clustering through a careful selection of research papers, covering current and related concepts and methodologies while identifying key challenges and recent developments in this area. The book has five clear sections, basics, visualization, algorithmic and computational aspects, real-time and dynamic clustering, applications and case studies, the computational extension of fuzzy clustering, and its effectiveness and distribution in handling high-dimensional problems, distributed problem solving, and uncertainty management. The book presented the important and relevant phases of cluster design including the role of fine-grained information, fuzzy sets to realize the human-centric aspect of data

analysis, system-modeling demonstrations of how the results can facilitate more detailed development of the model and improve interpretation, fuzzy clustering-centric applications, and case studies [57].

A given collection of data is divided into a certain number of clusters by clustering algorithms (groups, subsets, or categories). Although there isn't a single definition that applies to all clustering algorithms, there are as many alternative clustering approaches as there are clustering algorithms themselves (B. Everitt, S. Landau, and M. Leese, 2001). The majority of researchers (Chapman & Hall, 1999; Hansen P., and Jaumard B., in 1997; Jain A. and Dubes R., in 1988) defined clusters as having internal uniformity and external separation, which states that patterns within the same cluster must be similar to one another but not to patterns within other clusters.

Fung, G. offered a thorough discussion of the most widely used basic clustering algorithms as well as an overview of the most fundamental approaches. Parametric and nonparametric clustering approaches were each given their category in the paper. The Parametric algorithms include generative or probability-based models, Gaussian mixture models, C-Means fuzzy clustering, reconstruction models, K-means and K-median, and deterministic annealing, while the non-parametric methods include hierarchical clustering algorithms. The researcher also conducted numerical tests on some algorithms using various datasets with several initial cluster points [58].

Xu, R. & D. Wunsch for datasets used in statistics, computer science, and machine learning, surveyed clustering techniques. The study examined the numerous methods published in the literature while concentrating on clustering algorithms. As they originate from various academic areas and seek to address various issues, these algorithms each have their advantages and disadvantages. The article discussed the traveling salesman problem, several benchmark datasets, and applications in bioinformatics, a relatively new field that is now drawing a lot of attention. The study also investigated many open issues regarding clustering and clustering algorithms due to the many inherent uncertainties that exist despite the success of clustering algorithms in various applications. These issues have already and will continue to encourage intense efforts by a wide range of disciplines [59].

In his study, Abbas, O.A. reviewed and contrasted a number of data clustering algorithms. The methods up for comparison include hierarchical clustering, expected value maximization, partition-based clustering, and neural network-based clustering algorithms. The size of the dataset, the number of clusters, the kind of dataset, and the kind of software program employed are taken into consideration while comparing algorithms. The results are based on how well the clustering method performs, produces high-quality results, and is accurate [60].

Tian, Y., and D. Xu Beginning with a definition of clustering and consideration of its fundamental components, including distance and similarity measurements and scoring indications, the review paper's authors then examined the clustering method from both traditional and current

viewpoints. The old approach uses nine categories and 26 algorithms, while the modern approach uses 10 categories and 45 algorithms. The major goal of the paper was to introduce the fundamental concepts behind frequently employed clustering algorithms, identify each source, and evaluate each algorithm's advantages and disadvantages. The most practical and well-researched 19 types of regularly used clustering algorithms were chosen [61].

Rodriguez, M.Z., Comin, C.H., Amancio D.R., Casanova, D., Bruno O.M, Rodrigues, F.A and Costa, L.D.F., are among the authors of the study that performed a comparison of clustering algorithms. The authors assumed a normal distribution of the data and conducted a systematic comparison of nine well-known clustering techniques implemented in the R language. A thorough technique for producing a wide range of heterogeneous datasets with well specified qualities, such as distances between classes and correlations between features, was the main focus of the task. The researchers created a performance comparison of nine popular clustering approaches using the R language package and applied it to 400 synthetic datasets. The article assessed the clustering algorithm's sensitivity to parameter configuration as well. Standard parameter, single parameter variation, and random parameter variation were all simulated. In order to attain various performances in other circumstances, the research study underlined that all of the results stated in the paper pertain to the specific configuration of regularly dispersed data and the implementation of the algorithm. The study's findings demonstrated that, given the typical setup of the algorithms employed, spectral clustering methods frequently perform very well. The study also found that the default configuration of the selected implementations isn't always accurate and that in these situations, a straightforward method based on a haphazard choice of parameter values works well as an alternative to boost performance. The different trials for the clustering methods under consideration yielded several fascinating results in addition to providing useful advice on deploying clustering techniques [62].

Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Alshatri, N., Tari, Z., Fahad, A., and Bouras, A., conducted a thorough analysis of the clustering techniques put forward in the literature in their study. The study was intended to show the future direction of new algorithm development and guide the selection of big data clustering algorithms. The article proposed a categorization framework for classifying a set of clustering algorithms. The classification framework was developed from a theoretical point of view, which would automatically recommend the best algorithm to network experts, but hide all technical details that are not relevant to an application. Categories are hierarchy-based, density-based, partitional-based, model-based, and grid-based. Typical algorithms are BIRCH, fuzzy c-means, expected value maximization, DENCLUE, or optimal grid. This means that future clustering algorithms can also be integrated into the framework according to the proposed criteria and characteristics. In addition, the most representative clustering algorithms in each category are empirically analyzed using various metrics and traffic datasets [63].

The research paper written by Patibandla RL., and Veeranjaneyulu N., explored the use of the same techniques in distributed systems and Hadoop platforms, and the exploration of several existing clustering techniques suited for huge, unstructured, and semi-structured data. Unstructured data was described in the article as data from CCTV systems, social networking sites, and other resources that produce a lot of data. The white paper discussed parallel distributed computer systems, which are utilized for tackling complex issues, and clustering methods, which are useful for unstructured and semi-structured data. Expected value maximization, parallel k-means methods, and Map Reduce are some of the algorithms covered [64].

Data points are divided into k divisions by partition clustering techniques, with each partition denoting a cluster. The division is based on a certain specific objective function. One such reference function is the minimization of square error criteria. Elavarasi, S.A., Akilandeswari, J., and Sathiyabhama, B., gave an overview of various partition-clustering algorithms in their research work. This article discussed the methods utilized in these approaches, the parameters that determine their performance, and the overall effectiveness of partition clustering algorithms. The scaling technique, the k-means partition-clustering algorithm, and its variations were all covered in the white paper. The drawbacks of the k-means algorithm, which determines the ideal k-value and initial centroid for each cluster, were also discussed in this work. According to the review report, this flaw is fixed by using ideas like genetic algorithms, simulated-annealing, harmonized search strategies, and ant colony optimization [65].

T. Velmurugan and Santhanam, T. (2011) conducted an experimental study of partition-based clustering techniques in data mining. In addition to identifying partition-based algorithms like the k-means algorithm, k-medoids, and fuzzy c-means algorithm, the study also described the partition technique of clustering. The study's findings demonstrated that small to medium-sized datasets can be efficiently searched for spherical clusters using partition-based clustering techniques. The study also found that the k-means clustering algorithm's benefit is its speedy execution, but its disadvantage is that the user must know the number of potential clusters in advance. While the k-medoids approach performed better for large datasets, the k-means algorithm generally functioned well for small datasets. The fuzzy c-means algorithm seems to perform an intermediary between the other algorithms with higher execution times [66].

S. Na, L. Xumin, and G. Yong showed in their research study a quick and effective approach to cluster data points. The approach suggested in the article ensures that the full clustering procedure is completed in O (nk) time without compromising the accuracy of the clusters. This is accomplished by asking a straightforward data structure to request storing some data in each iteration that will be needed in the following iteration. The revised method reduces execution time by avoiding repeated estimates of the distance between each data object and the cluster's center. According to experimental findings, the revised technique can efficiently increase clustering's

speed and accuracy while also lowering the computing complexity of the k-means clustering algorithm [67].

Taking into account the drawbacks of conventional k-means clustering techniques, Wang, J., and Su, X introduced an enhanced k-means method that makes advantage of noise data filters in their article. The k-means clustering algorithm, which is a partition-based cluster analysis technique, was introduced by Mac Queen in 1967, according to the study. The k-means technique is frequently employed in cluster analysis because it is more effective, scalable, and quickly converges when processing huge datasets, according to the article. The algorithm does, however, have a lot of flaws. The process is impacted by noise points, the number of clusters K needs to be established, and the initial cluster's center was randomly selected. The scientists' new, improved approach used a density-based detection technique based on the properties of noise data. The approach extends the original algorithm by including steps for noise data detection and processing. The cluster aggregation of the clustering results is greatly enhanced, the impact of the noise data on the k-means algorithm is effectively decreased, and the clustering results are better by preparing the data before clustering the dataset and avoiding this noise data [68].

Mohamad, I. B., and Usman, D.A. developed a new method of k-means clustering that uses standardized techniques to generate optimal quality clusters. Extensive experiments with infectious disease datasets were conducted to examine the effectiveness of standardization and to compare the effectiveness of three different standardization methods in traditional k-means clustering algorithms. The results show that prior standardization of the clustering algorithm leads to better, more efficient, and more accurate cluster results. The study reiterated that it is also important to choose a specific standardization method, depending on the type of dataset that is being analyzed. In addition, by comparing the results of the infectious disease dataset, the results obtained by the z-score standardization method proved to be more effective and efficient than the min-max and decimal scaling standardization method utilizing the k-means clustering algorithm [69].

T M Kodinariya & P.R. Makwana discussed how to choose the appropriate number of clusters, one of the most contentious issues in k-means clustering. Because "clusters are typically in the eyes of the observer, not the facts," some scholars believe it to be untrue. The review paper uses the k-means method to attempt to solve the cluster number selection problem. The end-user is typically required to supply the number of clusters in advance by the k-means clustering method, but this is not practical for end-users as it necessitates domain knowledge of all records and datasets. In order to estimate the number of clusters in a dataset that has been published in the literature, the study detailed six distinct methodologies, including statistical indicators, variance-based methods, information theory, goodness-of-fit methods, and so on [70].

A brand-new partitioning approach known as partitioning around medoids was introduced by Kaufman L. and Rousseeuw P.J. in their book "identifying groups in data: an introduction to

cluster analysis" (program pam). The PAM program's method is built around looking for k representative objects in the dataset. These objects must represent various facets of the data structure, as their name suggests. Such representative objects are commonly referred to as Centro types in the cluster analysis literature. The so-called medoid of a cluster is a typical object in the PAM algorithm. When k representative objects have been found, k clusters are created by linking each object in the dataset to the following representative object [71].

The k-medoid partitioning method described above gives very satisfactory results in many situations. However, as can be deduced from the time and memory requirements of the PAM program, the PAM program is not practical for clustering large datasets. For this reason, a new process was designed specifically for large applications in the book written by Kaufman L, and Rousseeuw P.J. The method is also based on the k-medoid approach and has been implemented in the CLARA program (abbreviated as Clustering LARge Applications). Clustering a large number of objects using CLARA is a two-step process. First, the sample is extracted from the set of objects and clustered into k subsets using the k-medoid method. This is also a representative of k objects (which run with the same algorithm as PAM). Each object that does not belong to the sample is then assigned to the closest of the k representative objects. This clusters the entire dataset. A measure of the quality of this clustering is obtained by calculating the average distance between each object in the dataset and its representative objects. Five samples are extracted, clustered, and then the sample with the smallest average distance is selected. The resulting clustering of the entire dataset is then further analyzed. For each cluster, CLARA reports its size and medoids and prints a complete list of its objects [71].

Han, J., and Ng, R. T. proposed a new clustering technique called CLARANS (Clustering of Large Applications based on RANdomized Search) (Clustering of Large Applications based on RANdomized Search). Its goal is to locate any potential spatial structures in the data. According to experimental findings, CLARANS outperforms other clustering techniques in terms of efficiency and effectiveness. The essay explained how CLARANS effectively manages both point and polygon objects. Convex and non-convex polygon objects can be clustered together very effectively using one of the methods under consideration, the IR approximation. The research article created two spatial data mining algorithms based on CLARANS with the goal of establishing the connection between spatial and non-spatial attributes. Both approaches can unearth information that is challenging to find with the current geographical data mining tools. Compared to CLARA, it has the advantage that the search space is not limited to a specific preselected subgraph, as in CLARA. As a result, CLARANS can produce clustering at the same runtime with much higher quality than that produced by CLARA [72].

Dabhi, D. P. and Patel, M.R. provided a detailed review and summary of various types of hierarchical clustering techniques. The white paper outlines the various hierarchical clustering

techniques used in data mining. The article provided a quick rehearsal of the various hierarchical clustering techniques used for data mining by including definitions, procedures, how the clustering techniques work, etc. The paper also gave a detailed classification of hierarchical clustering techniques and their respective algorithms with the advantages, disadvantages, and comparative study of all the algorithms. The hierarchical clustering methods discussed are divisive (top-down) and agglomerative (bottom-up). The algorithms discussed include complete linkage, single linkage, agglomerative nesting (AGNES), average linkage, clustering using representatives (CURE), RObust Clustering using linKs (ROCK), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Clustering Using Dynamic Modeling (CHAMELEON), and Monothetic Analysis (MONA) [73].

The book “handbook of cluster analysis by Marina Meila, Christian Hennig, Pedro Contreras, and Fionn Murtagh” looked at both traditional agglomerative hierarchical clustering, and its recent developments in the grid- or cell-based approaches. The book described various algorithmic aspects such as well-definedness (such as inversion) and computational characteristics. This includes the latest agglomerative hierarchical clustering algorithm, which uses a nearest-neighbor chain and reciprocal nearest neighbors. Finally, the book described some standard R commands and how to get a handy display [74].

Unfortunately, many traditional agglomerative clustering algorithms are not known to be robust to noise. Balcan, M.F., Liang, Y., and Gupta, P., proposed and analyzed a new robust algorithm for bottom-up aggregate clustering. The authors showed that if the data meets many natural properties and the traditional agglomerative algorithm fails, the algorithm could be used to cluster it accurately. The research also showed how to adapt the algorithm to an inductive setting where the given data is just a small random sample of the entire dataset. Various experimental evaluations of the real world and synthetic datasets showed that the proposed algorithm is superior to other hierarchical algorithms in the presence of noise [75].

Chris Ding and Xiaofeng introduced the process of merging and splitting in a hierarchical clustering technique. They provided a thorough analysis of selection methods and suggested some new ways to determine the best way to select the next cluster for split or merge on a cluster. The authors conducted extensive clustering experiments to test the eight selection methods and found out that the averaged similarity is the best method for divisive clustering, and the Min-Max linkage is the best for agglomerative clustering. Cluster balance was a key factor in achieving good performance there. The article also introduced the concepts of objective function saturation and clustering target distances to effectively evaluate the quality of clustering [76].

Rani, Y. & Rohil, H. described some of the enhanced hierarchical clustering techniques in their research work. The authors have provided additional criteria that make it possible to select the best algorithms from the list. The authors propose that the hierarchical clustering algorithm can be

enhanced by combining it with additional methods. Because pure hierarchical clustering approaches cannot be modified after a merge or split decision has been made, this is important to improve their quality. If the merge or split option is not appropriately chosen at a specific step, it may cause a modest reduction in cluster quality [77].

When determining the true clustering structure of a dataset, hierarchical clustering algorithms typically outperform partitioning approaches. However, the hierarchical clustering algorithm just computes the dataset's hierarchical representation; it does not really produce a cluster. Because of this, it is inappropriate for use as an automatic pre-processing step for other algorithms running on the acknowledged cluster. This holds for both reachability plots and dendrograms, which have different advantages and disadvantages as proposed methods for representing hierarchical clusters. The authors of the study by Sander, J., Qin, X., Lu, Z., Niu, and Kovarsky explored the connection between the dendrogram and the reachability plot and showed how they can be transformed into one another, demonstrating that they fundamentally carry the same information. On the basis of the reachability plot and dendrograms, the study then proposed and assessed a new method for automatically identifying the critical clusters in a hierarchical cluster representation. As a result, hierarchical clustering can now be used as an automatic preprocessing step that doesn't require user input to choose a cluster from a hierarchical cluster representation [78].

Salvador, S. and Chan, P., proposed an efficient algorithm, the L-Method, which finds the "knee" in the "Number of Clusters versus Clustering Scoring Metrics" graph. The authors elaborated on the L method, which has been shown to work very well in determining the optimal number of clusters or segments in a segmentation / hierarchical clustering algorithm. Hierarchical algorithms with greedy evaluation metrics are especially effective. In the evaluation, L-Method was able to determine the optimal number of segments in 10 out of 11 cases for the greedy hierarchical segmentation algorithm and 10 out of 12 cases for the hierarchical clustering algorithm. Algorithms that use global rating metrics did not work well with the L-method because the knees are not noticeable in the evaluation graph. Gap statistics and permutation tests were also evaluated, and the L-method achieved much better results in the evaluation. The L-method is much more efficient than both gap statistics and permutation tests since it takes only a moment to determine the number of clusters, not minutes or hours. The paper reiterated that iterative improvement of the knee is a very important part of the L-method. The iterative improvement algorithm described in the paper ensures that the L-Method always runs under optimal conditions with balanced lines displayed on both sides of the knee, regardless of the size of the evaluation graph or the position of the knee [79].

Motivated by the fact that most of the work on hierarchical clustering is based on the provision of algorithms rather than optimizing specific goals, Dasgupta embraced similarity-based hierarchical clustering as a combinatorial optimization problem where a “good” hierarchical

clustering minimizes a particular cost function. He showed that this cost function has certain desirable properties. To achieve the optimal cost, individual components (that is, different items) must be separated at a higher-level in the hierarchy, and if the similarities between the data items are the same, all clustering achieves the same cost [80].

CohenAddad, V., Kanade, V., MallmannTrenn, F., And Mathieu, C., adopted an axiomatic approach to defining "good" objective functions for both similarity and dissimilarity-based hierarchical clustering. They characterized a set of executable objective functions with the property that ground truth clustering has optimal values if the input allows "natural" hierarchical ground truth clustering. The authors showed that this set includes the objective function introduced by Dasgupta. Equipped with a suitable objective function, the paper analyzed the performance of practical algorithms, as well as developed better and faster algorithms for hierarchical clustering. The article also initiated a beyond worst-case analysis of the complex problem and designed algorithms for such scenarios [81].

Traditional density-based clustering algorithms and new density-based clustering algorithms were compared by Loh, W.K., and Park, Y.H. A cluster of dense items is created through density-based clustering and is divided by sparse regions. As representative examples of conventional density-based clustering algorithms, DBSCAN, OPTICS, and DENCLUE were discussed. Recent developments in density-based algorithms have led to the usage of multi-computer and multi-core processors to analyze massive data sets and replace traditional density-based algorithms that use a single thread on a single computer. In such algorithms, some concerns with parallel processing must be taken into account. It is necessary, in particular, to offer a whole dataset in slices and suggest effective strategies to reduce the data and manage transmission. The study also suggested a number of novel methods to enhance the functionality of well-known clustering algorithms including GSCAN, PDBSCAN, and CUDADClust. All of these approaches maximize multi-core CPUs and GPUs [82].

Only a limited number of methods are covered by the existing research on density-based clustering algorithms (DBCLAs). These studies don't offer full details on the numerous DBCLAs that have been previously proposed, like algorithm classifications. A thorough analysis of the numerous DBCLAs during the previous 20 years (1996 to the present) and their classification was provided by Bhattacharjee, P., and Mitra, P. The DBCLAs are classified into many classes under each of the following four categories: density definition, parameter sensitivity, execution method, and data type. The comparison of DBCLAs also takes into account the similarities, differences, and conceptual dependencies among them. In addition, the study revealed a number of applications for DBCLA in fields like astronomy, geosciences, molecular biology, geography, and multimedia [83].

A comparison of the three density-based clustering methods DENCLUE, DBCLASD, and DBSCAN was provided by Nagpal, P.B., and P.A. Mann. For their comparison, six parameters are

taken into account: complexity, cluster shape, input parameters, noise handling, cluster quality, and run time. The results of the study are supported by solid experimental evaluation, and the table of results shows that the DENCLUE algorithm has the lowest execution time while the DENCLUE has the highest run time. When it comes to cluster quality, DBCLASD takes the lead while DENCLUE has the lowest quality. The analysis of the results in this paper serves as a guide to finding the right density-based clustering algorithm in different situations [84].

This study examined many common clustering algorithms for density-based clustering on data streams. It was conducted by Amini A, Wah TY, and Saboohi H. The white paper's key benefit is that it gave a thorough review of scoring metrics and density-based data stream grouping techniques. The survey also separated the algorithms into the micro-cluster and grid algorithms, two fundamental taxonomies, making it simpler to investigate density-based clustering algorithms. The research study provides an overview of the most significant density-based clustering methods for data streams, discusses their special features and drawbacks, and offers solutions to common problems with data stream clustering. Additionally, the assessment measures that were utilized to check the cluster's quality and assess the algorithm's performance were discussed [85].

In this article, Chen, Y. and Tu, L., proposed D-Stream, a new framework for clustering stream data using a density-based approach. The algorithm maps all input data to grids calculates the density of each grid and uses a density-based algorithm to group the grids. In contrast to previous algorithms based on k-means, the proposed algorithm has an online component that maps each input dataset to a grid and an offline one that calculates the grid density and clusters the grid based on the grouped densities. The algorithm uses a density attenuation method to capture changes dynamically in the data stream, taking advantage of the complex relationships between decay coefficients, data densities, and cluster structures, to efficiently and effectively generate and adapt clusters in real time. In addition, sophisticated and theoretically appropriate techniques are developed to detect and remove sporadic grids and dramatically improve spatial and time efficiency without affecting clustering results. This technology enables high-speed data stream clustering without compromising the quality of clustering [86].

Based on the specifications for spatial data clustering algorithms, Paris Mara M., Lopez D., and Centil Kumar NC., offered a thorough overview of five density-based clustering methods, including VDBSCAN, DBSCAN, ST-DBSCAN, DVBSCAN, and DBCLASD. Every algorithm has distinctive qualities. The authors of this research examined the method in terms of the factors required for the generation of significant clusters. As a consequence of the investigation, a comparison of the input parameters, cluster shape, density, and data type was carried out and tabulated [87].

DBSCAN, a clustering technique based on the idea of cluster density, was introduced in this article by Ester, M., Kriegel, H.P., Sander, and Xu. The method aids the user in selecting the right

value for the only input parameter it requires. The SEQUOIA 2000 benchmark's real and synthetic data were used to evaluate performance. The outcomes of these tests demonstrate that DBSCAN is substantially better at detecting clusters of any shape than the well-known CLARANS method. Additionally, research has demonstrated that DBSCAN is at least 100 times more effective than CLARANS [88].

The density-based algorithm DBSCAN was generalized in two significant ways by the clustering algorithm GDBSCAN (Ester et al., 1996), which was introduced in this article by Sander, Ester, Kriegel, and Xu. Both point objects and spatially expanded objects can be grouped using both spatial and non-spatial properties using GDBSCAN. The paper presented the general concepts of density connection sets and the algorithms for their discovery after reviewing the relevant work. Performance assessments, both analytical and experimental, demonstrated the effectiveness and efficiency of GDBSCAN in large spatial databases. Four applications were also introduced by the authors. 2D points in astronomy, 2D polygons in geography, 3D points in biology, and 5D points in geography demonstrate the applicability of GDBSCAN to practical issues [89].

A theoretically and practically enhanced density-based hierarchical clustering method was put forth by Campello, Moulavi, and Sander. It offers a clustering hierarchy that creates a condensed tree of critical clusters. The newly introduced density-based clustering approach (HDBCSAN) offers a full density-based cluster hierarchy that represents all potential DBSCAN-like solutions for an infinite range of density thresholds. From this hierarchy, a flat partition made up of clusters that were extracted from the best local cut through the cluster tree, and a simplified tree of critical clusters can be created. Extensive experimental evaluation of numerous real-world datasets revealed that, for a variety of real-world data, it outperforms current state-of-the-art density-based clustering algorithms [90].

Using examples from sensor databases, spatiotemporal applications, and biometric information systems, Kriegel, H. P., and Pfeifle, M., showed how to create hierarchical density-based grouping from ambiguous and uncertain information. The authors provided the theoretical foundation for hierarchical density-based clustering of uncertain data as well as examples of practical applications of these ideas. The resulting FOPTICS algorithm can be used to effectively and efficiently cluster ambiguous data, such as moving objects. This algorithm follows the general paradigm of integrating fuzzy-distance functions directly into the data mining algorithms instead of processing lossy aggregated information. Experimental evaluation of the newly introduced FOPTICS clustering algorithm showed that it achieves far more accurate results than current comparison partners without compromising efficiency [91].

Baraldi, A. and Blonde, P. from existing literature, derived a unique interpretation of inductive learning using fuzzy clustering, intended as a synonym for soft and competitive parameter adaptation in clustering systems. This conceptual equivalence is used as an integrated framework

when comparing clustering algorithms. The purpose is to propose a theoretical framework that can compare the expressiveness of a clustering system based on a set of meaningful common functional characteristics. This article covers the following topics related to the clustering approach found in the literature: Relative (stochastic) and absolute (possibilistic) fuzzy membership functions related to Bayesian rules, batch and online learning, prototypes editing schemes, growth and pruning networks, modular network architectures, topologically complete mappings, ecological networks, and neuro-fuzziness. The framework selected a set of basic functional features to use as dictionary entries when comparing clustering models [92].

In part II of their article, Baraldi, A., and Blonde, P. studied five clustering algorithms that were picked from the literature and assessed and compared them based on the chosen properties of interest described in the prior study. Self-organizing maps underlie these clustering models (SOMs). Fuzzy Adaptive Resonance Theory (Fuzzy ART), Growing Nerve Agents (GNG), Fully Self-Organized, and Fuzzy Learning Vector Quantization (FLVQ) simplified the Adaptive Resonance Theory (FOSART). Comparing the positive values of these algorithms revealed that while Fuzzy-ART does not develop soft competitive adaptation methods, SOM, FLVQ, GNG, and FOSART do. Ironically, only FLVQ and FOSART apply fuzzy set-theoretical ideas among these four fuzzy clustering systems (such as absolute and relative fuzzy membership functions). For FLVQ, this is susceptible to asymptotic errors. In conclusion, the only successful fuzzy clustering algorithms are GNG, SOM, and FOSART [93].

Bataineh, K.M., Naji, M., and Saqer, M., reviewed and compared the two most common fuzzy clustering techniques, the Fuzzy C-mean (FCM) algorithm, and the subtractive clustering algorithm. The comparison is based on a validity measurement of the clustering results. A non-linear function is modeled and the two algorithms are compared depending on their modeling ability. The performance of the two algorithms was also tested based on experimental data. The authors provided a brief review of the literature and showed the basic parameters that control each algorithm. The clustering results of each algorithm were evaluated using the Dave, Bezdek, and Xi-Beni validation metrics. For most of the systems described earlier, the authors discovered that increasing the number of clusters generated improved the validity index value and optimal modeling results were obtained when the validity index is at the optimal value. In addition, models generated from subtractive clustering are usually more accurate than models generated using the FCM algorithm. Further analysis of the FCM algorithm demonstrated that a training algorithm is required to generate a model accurately using FCM. However, for subtractive clustering, no training algorithm is required. The study showed that FCM has an inconsistency problem, as the algorithm chooses an arbitrary υ matrix each time, so different FCM runs will produce different results. Subtractive algorithms, on the other hand, produce consistent results [94].

In this review paper, Gosain, A. and Dahiya, S., conducted a comprehensive study and experiment based analysis of the performance of all major fuzzy clustering algorithms namely the Possibilistic C-Means Algorithm (PCM), Fuzzy C-Means Algorithm (FCM), FCM-σ, T2FCM, KT2FCM, IFCM, KIFCM. , IFCM-σ, KIFCM-σ, NC, CFCM, Possibilistic Fuzzy C-Means Algorithm (PFCM), and DOFCM. The authors experimented with standard data sets (D12) in the presents of noise and outliers to determine the analysis of their performance. The study observed that PCM and PFCM improved the performance of FCM in presence of noise but not significantly. T2FCM and KT2FCM significantly improved clustering results over FCM, PCM, PFCM, and KT2FCM are even more robust than T2FCM in presence of noise; IFCM-σ significantly improves centroid position over FCM and IFCM; NC and CFCM suppressed the effects of outliers and proved to be the best robust algorithms when noise and outliers are introduced. DOFCM worked best to first accurately exactly identify outliers and then provide the best centroids compared to all other techniques and algorithms [95].

The authors of this publication, Bezdek, J.C., Ehrlich, and Full, communicated FORTRAN IV coding for fuzzy c-means (FCM) clustering applications. Numerous issues involving the analysis of geo-statistical data can be solved with the FCM application. For each set of numerical data, the computer program creates fuzzy partitions and prototypes. The assessment of existing substructures and the suggestion of new substructures for unknown data are both facilitated by these partitions. The generalized least squares objective function is the clustering criterion used to combine the subsets. This program's features include the ability to select between three criteria (Euclidean, Mahalanobis, or diagonal), a weighting factor that may be adjusted to control noise sensitivity, the acceptance of a variable number of clusters, and outputs that contain various measures of cluster validity [96].

Miyamoto, S., Honda, K., and Ichihashi, H.., wrote a book titled algorithms for fuzzy clustering. The main subject of the book is the fuzzy c-means proposed by Dan and Bezdeck and their variations, including recent studies. The book focused on fuzzy c-means because most fuzzy clustering methodologies and application studies use fuzzy c-means. In contrast to most studies of fuzzy c-means, the focus of this book is on a family of algorithms that use entropy or entropy-normalized methods. In the book, one of the intentions of the authors is to clarify the theoretical and methodological differences between the traditional Dunn and Bezdek method and the entropy-based method. The authors do not claim that the entropy-based method is superior to the traditional method but believe that the fuzzy c-means method can be complemented by adding the entropy-based method to the Dunn and Bezdek method. This is to deepen the nature of both methods, by contrasting the two algorithms [97].

In their paper, Gath, I., and Geva, A.B. presented how to carry out fuzzy classifications without making any assumptions in advance on the number of clusters in a dataset. Based on

performance evaluations using hyper-volume and density criteria, the cluster validity assessment is made. The fuzzy maximum likelihood estimation algorithm and the fuzzy K-means algorithm were combined to create the new algorithm (FMLE). The UFPONC (Unsupervised Fuzzy Partition Optimal Class Amount) approach performs well when the size, density, and number of data points inside each cluster vary widely. On several simulated and real datasets made from sleep EEG signals, the authors tested the technique [98].

Self-organizing maps (SOMs) neural networks, also known as Kohonen neural networks, are an effective tool for analyzing multidimensional data. This network can be used for cluster analysis while preserving the data structure (topology), so similar inputs (data) remain close to each other at the output layer of the network. Ghaseminezhad, M.H., and Karami, A., in their paper introduced a new SOM-based algorithm that can automatically cluster discrete groups of data using the unsupervised method. The process consists of three phases. In the first phase, an algorithm called "Second Winner" is executed; in which neurons in the competing layer of the network find their original location in network space. The second phase uses a method called "batch learning", in which the training of the SOM network is completed at the end of this phase. Finally, in the third phase, data clustering is completed by removing the incorrect connections between neurons. The study demonstrated the accuracy and effectiveness of the proposed approach using three real-world datasets, including iris, wine, and glass datasets, an example of non-linearly separable synthetic data. The innovative SOM algorithm works much better due to its flexible structure and the use of batch learning processes in clustering discontinuous data [99].

By utilizing an incremental self-organizing neural network to learn the data distribution of each cluster, Liu, H., and Ban, X.J. proposed a novel method for cluster recognition. The suggested method builds a dynamic 2D topology network automatically that is used to analyze both cluster data and specific information for each cluster. The number of clusters is not a requirement for the suggested method as a clustering methodology. By counting the number of components in the created topology graph, users can quickly determine the number of clusters. The suggested procedure can be applied in stages, and any format can be used to display the identified clusters. The suggested method was tested using one original hand gesture dataset, five artificial datasets with various data distributions, and five artificial datasets. The reported experimental findings demonstrate the good performance of the suggested algorithm in controlling outliers, robust learning, efficiency, and showing data linkages [100].

Chang, C.C. and Chen, S.H., presented a comparative study of artificial neural network-based two-stage clustering. Artificial neural networks (ANNs), which can eliminate noise and reduce data complexity, are considered one of the best mediators in a two-step clustering process. Various ANN-based two-step clustering techniques have been proposed individually such as the hybrid of k-means and self-organizing feature map (SOFM), or SOFPM + k-means, adaptive

resonance theory (ART) neural network + k-means, and Fuzzy ART + k-means. However, the performance of these methods has not yet been investigated. In this study, a preliminary comparative analysis is performed on four benchmark datasets and a real market dataset is used to simulate different conditions for evaluation purposes. Experimental results suggest that a highly accurate self-organizing feature map can potentially improve the effectiveness of decision-making [101].

Han, M. and Xi, J., in their article described how to train a new feedforward neural network, the Radial Basic Perceptron (RBP) neural network, to distinguish between different sets in the R^L output vector. RBP Neural network is based on a neural network with a radial basis function (RBF) and a neural perceptron network. The node is not fully connected, but two hidden layers use selective connections. Training algorithms corresponding to the structure of the RBP network are presented in the study. It uses the Input / Output Clustering (IOC) method to provide an efficient and powerful method for building RBP networks that are very easy to generalize. First, during the learning process, the RBP neural network uses the IOC method to define the number of hidden layer units and select the cluster center. Second, the center width parameter is self-adjustable according to the information contained in the training sample. The effectiveness of this network is illustrated using an example application for component analysis of building materials. Simulations show that RBP neural networks can be used to successfully predict building material components and achieve good generalization [102].

The growth of digital data necessitates the creation of scalable solutions for data organization in a useful and approachable manner. However, because they make the assumption that the data can be split linearly, many clustering methods created to handle huge datasets perform poorly with real datasets. Kernel based clustering algorithms may capture non-linear data structures, but when the number of clustering objects approaches tens of thousands, they become too slow and demand too much storage. For big datasets, an efficient approximation of the kernel k-means algorithm was proposed by Chitta, R., Jin, R., Havens, T.C., and Jain, A.K. By restricting the cluster's center to a small subspace that covers a collection of randomly sampled data points, it is possible to avoid having to calculate the complete kernel matrix. The suggested technique's efficiency in terms of computing cost and memory needs was shown theoretically and empirically by the authors, who also showed that a full kernel matrix could be utilized to provide clustering outcomes comparable to those of the kernel k-means algorithm [103].

In an article they published, Zhao, B., Kwok, J.T., and Zhang, C., extended multiple kernel learning to include unsupervised learning. The authors specifically suggested a multi-kernel clustering algorithm (MKC) that simultaneously finds the best cluster labeling, the largest margin hyperplane, and the best kernel. Maximum Margin Clustering (MMC) has recently drawn a lot of attention from the machine learning and data mining fields. Finding the hyperplane with the largest

margin for each potential cluster label is how clustering is accomplished when the data is first projected into the kernel-induced feature space. Choosing the appropriate kernel function is crucial for effective maximum margin clustering, as it is with any kernel technique. Results from experiments on both toy and actual datasets demonstrate the algorithm's usefulness and efficiency [104].

For extending the original neural gas (NG) algorithm into a higher-dimensional feature space, Qin AK. and Suganthan PN. presented the kernel neural gas (KNG) approach in their article. The clustering of non-linear structured datasets was effectively addressed by the authors' kernel neural gas (KNG) approach. The KNG approach is less susceptible to initialization than some kernel clustering algorithms since it makes use of neighborhood coordination and sequential learning techniques. The imbalanced clustering issue has also been addressed using the distortion-sensitive KNG method. Results from experiments show how well the KNG algorithm performs. The study did find, however, that the choice of kernel parameters affects how well kernel algorithms cluster data. As a result, proper parameter tuning was carried out to get the optimal clusters [105].

The "kernel technique" is frequently used in contemporary machine learning communities to create non-linear adaptations of linear methods. Examples of such efforts include the support vector machine (SVM), kernel principal component analysis, kernel Fisher discriminant analysis, and contemporary kernel clustering techniques. The data is often initially mapped to a potentially much higher feature space via nonlinear mapping in unsupervised clustering methods that employ the kernel method before clustering. Unless extra projection approximations from the feature to the data space are utilized, as discussed in the existing literature, the clustering prototype in these kernel-clustering techniques is in a high dimensional feature space, making it ambiguous and counterintuitive. In their research work, Zhang, D.Q., and Chen, S.C., proposed the kernel fuzzy c-means algorithm, a new clustering technique based on the conventional fuzzy clustering algorithm (FCM) (KFCM). The new kernel-guided metric in KFCM replaces the original Euclidean norm metric in FCM in the data space. The clustering outcome can be reformulated and evaluated in the original space because the clustered prototype is still present in the data space. The research investigation shows that KFCM is resilient to noise and outliers and can survive clusters of different sizes. Finally, partial data are clustered using this attribute. KFCM has improved clustering performance and greater robustness than some changes in FCM, according to experiments with two synthetic datasets and one real dataset [106].

A multi-kernel fuzzy c-means algorithm (MKFC) was suggested by Huang HC, Chuang YY, and Chen CS. It extends the fuzzy c-means algorithm in a multi-kernel learning configuration. A popular soft clustering method is fuzzy c-means, however, its usefulness is mostly restricted to spherical clusters. The kernel fuzzy c-means technique uses kernel trickery to translate data with

non-linear correlations to the appropriate feature space in an effort to solve this problem. Effective kernel clustering depends on the pairing or selection of kernels. Unfortunately, finding the ideal combination can be challenging in the majority of cases. MKFC is unaffected by inefficient kernels and unrelated features because of the integration of numerous kernels and automatically modifying kernel weights. The choice of kernels becomes less significant as a result. Additionally, it demonstrates that MKFC is a specific instance of multiple kernel k-means. The usefulness of the proposed MKFC algorithm has been demonstrated in experiments using synthetic and real data [107].

As the scope of cluster analysis grows and deepens, a special type of clustering algorithm named the spectral clustering algorithm raised great concerns among scientists. A recent development in the field of machine learning is the use of spectral clustering methods. Unlike traditional clustering algorithms, this can solve non-convex sphere non-convex spheres of sample spaces and it generalizes to a globally optimal solution. Ding, S., Zhang, L., and Zhang, Y., presented a survey of the current research status of spectral clustering algorithms and the principles and outlines of various application domains. First, the authors provided analysis and guidance of several spectral clustering algorithms from several aspects such as algorithmic ideas, key technologies, strengths, and weaknesses. In contrast, a few common spectral clustering techniques were chosen for examination and comparison. Last but not least, the analysis identified the principal issues and future directions [108].

Verma, D. and Meila, M., compared some of the popular spectral clustering algorithms from the following perspectives: displaying clustering quality using artificial and real datasets, implementing many variations of existing spectral algorithms, and comparing their performance to see which functions are more important. The purpose of the study was to compare the properties of many publicly available spectral clustering algorithms. Instead of determining which of the published algorithms is better, the authors wanted to evaluate which features would make spectral clustering more valuable. When clustering datasets, "quality" is in the eyes of the viewer, so clustering algorithms should be viewed not only as mutual competition but also as mutually complementary strengths and weaknesses. Therefore, the second goal of the research was to see how the different algorithmic approaches differ. The answer to the second question is almost negative. Theory predicts that the perfect S is the same for all three algorithms, and the results are strongly supported by experiments. All algorithms work very well when S is almost perfect and otherwise there is no clear winner. The multi-way spectral clustering algorithm has proved to be a bit more powerful, especially when the structure is easily found in the data. For recursive methods, the authors recommended using the N-cut measure over other methods, but otherwise, there is no clear winner [109].

Yan, D., Huang, L., and Jordan, M.I., proposed a general framework and introduced two fast algorithms for approximate spectral clustering. The algorithm uses local k-means clustering (KASP) and random projection trees (RASP) methods to pre-group adjacent points to produce a reduced set of representative points for spectral clustering. These algorithms significantly reduce the cost of matrix clustering for spectral clustering, while properly controlling the accuracy of clustering. Evaluation of a set of actual datasets shows that significant acceleration of spectral clustering can be achieved with little loss of clustering accuracy. In particular, our approximation algorithm allows you to perform spectral clustering on large datasets (poker hand datasets) consisting of 1 million instances on a single machine [110].

When it comes to finding clusters, spectral clustering techniques outperform the majority of conventional algorithms. However, when the size of the dataset is huge, spectral clustering has scalability concerns in terms of both memory utilization and computing time. Chen, W.Y., Song, Y., Bai, H., Lin, C.J., and Chan, E.Y. advocated parallelizing both memory use and computation on distributed computers to conduct clustering on huge datasets. Parallel methods can successfully minimize scalability concerns, as demonstrated by empirical investigations on big document datasets with 193,844 data instances and large photo datasets with 637,137 data instances. The study demonstrated the correctness and scalability of the spectral clustering algorithm's parallel implementation. There is no parallel algorithm that can defy Amdahl's law, although studies have demonstrated that the parallel spectral clustering technique can benefit from faster, higher-quality clustering performance the larger the dataset [111, 112].

Applying traditional clustering techniques to new types of data, such as streaming data and web/blog data, where the relationships between the data evolve poses new challenges. On one hand, a simple aggregation-based approach does not produce satisfactory cluster results due to long-term conceptual changes. On the other hand, short-term fluctuations occur often, which are caused by noise. It is desirable that the results of the cluster do not change dramatically in a short period, but should become smooth over time (temporal smoothness). Zhou, D., Song, X., Chi, Y., Tseng, B.L., and Hino, K., proposed two frameworks for incorporating temporal smoothness into evolutionary spectral clustering. Both frameworks define cost functions. This function introduces a second cost factor to adjust the smoothness over time, in addition to the traditional cluster quality cost function. Next, the (relaxed) optimal solution for solving the cost function is derived. It turns out that the solution is very intuitive and has a form similar to the traditional method of time series analysis. Experimental studies have shown that these new frameworks produce stable, consistent, and adaptive cluster results in the long term [113, 114].

Correlation spectral clustering is a new approach of spectral clustering that uses paired data based on kernel canonical correlation analysis to group images by defining pertinent paired data, such as text and video information. It was proposed by Blaschko, M.B., and Lampert, C.H. Finding

a non-linear projection of the image that correlates with the pertinent information is how this is accomplished. The suggested method employs a distinct similarity for every data representation, enabling the projection of previously unknown facts observed in just one representation (for example, images but not text). Spectral clustering can be applied to data with any number of modalities thanks to correlation spectral clustering. By applying a probabilistic interpretation of the CCA, the study demonstrated why correlated spectral clustering enhances modular spectral clustering and examined the effects of employing the empirical covariance matrix on the noise process. Compared to conventional spectral clustering, several publicly available datasets offer statistically significant empirical gains [115].

Elhamifar, E. and Vidal, R., investigated the problem of clustering a collection of data points in or near the union of a low-dimensional subspace. They proposed a subspace-clustering algorithm based on a sparse representation method called SSC. The algorithm searches the sparse representation of each point in the dictionary of other points, uses the sparse coefficient to create a similarity graph, and performs segmentation of the data by spectral clustering. The research showed that the algorithm succeeds in restoring the desired sparse representation of the data points under the appropriate conditions of subspace placement and data distribution. The main benefit of the algorithm is that in integrating the appropriate model into the sparse optimization program, it can directly handle data interference such as noise, sparse outliers, missing entries, and the more common class of affine subspaces. Experiments with real world data such as facial images and video movements have shown how effective the proposed algorithms are and their superiority over state-of-the-art [116].

Daumé, H., and Kumar, A., proposed a spectral clustering algorithm for multi-view settings that can access multiple views of the data. Each view can be used individually for clustering. The spectral clustering algorithm has a kind of co-training that is already widespread in semi-supervised learning. The true underlying clustering assumes that you assign points to the same cluster regardless of the view. Hence, the approach is limited to finding only matching clusters in all views. A common framework for the proposed algorithm is to learn clustering in one view and use it to "label" the data in another view thereby modifying the graph structure (similarity matrix). The adjustment of the graph is determined by the identifiable eigenvectors and is achieved by projections along with these directions. Our algorithm does not need to set hyperparameters, which is a big advantage in unsupervised learning. To demonstrate the superiority of the proposed algorithm, the study empirically compared some basic methods on real world and synthetic datasets [117].

Daumé, H., Rai, P., and Kumar, A., as part of spectral clustering, proposed another multi-view clustering approach called co-regularization. Co-regularization spectral clustering has joint optimization features common to spectral embedding in all views. The alternate maximization

framework reduces the problem to the goal of standard spectral clustering that can be solved efficiently with Eigen solvers. The proposed spectral clustering framework achieves this goal by co-regularizing the clustering hypothesis, using two co-regularization schemes to achieve this. Experimental comparisons with some baseline methods on two synthetic datasets and three real datasets demonstrate the effectiveness of the proposed approach [118].

The principles of natural selection and natural genetics serve as the foundation for the search and optimization technique known as the Genetic Algorithm (GA). In order to create search algorithms that are reliable and demand little problem information, several fundamental concepts from biology are artificially taken and applied. Using a genetic algorithm (GA) based clustering technique, Raghuwanshi, M.M., Sheikh, R.H., and Jaiswal published a survey article that highlights the state-of-the-art in this field. A genetic algorithm (GA) is applied to the clustering algorithm in order to enhance its performance. The most common evolutionary strategy is GA. GAs are used to provide adequate clustering and to evolve the right amount of clusters. The article discussed some currently used GA-based clustering methods and how they were applied to various issues and fields [119].

GA clustering is a genetic algorithm-based clustering method that was proposed by Bandyopadhyay, S., and Maulik, U. The algorithm employed the genetic algorithm's search function to locate the proper cluster center in the feature space and then applied a similarity metric to optimize the resulting cluster. The center of a predetermined number of clusters is encoded by chromosomes, which are represented as strings of real numbers. Using four synthetic and three actual datasets, the benefits of the GA clustering algorithm over the popular k-means technique are thoroughly illustrated [120].

In their publication, Garai, G., and Chaudhuri, B.B., developed a novel genetically controlled algorithm to address the clustering issue. Two steps make up the proposed genetic clustering algorithm. The initial dataset is divided into multiple fragmented clusters in the first phase, and the GA search process is dispersed across the space in the second phase. The Hierarchical Cluster Merge Algorithm is utilized in the second step (HCMA). Some of the fractured clusters are combined into a full k-cluster using the iterative genetic algorithm-based approach known as HCMA. The Adjacent Cluster Check Approach is a further element of this algorithm (ACCA). This method is used to determine whether two segmented clusters are close enough to one another to be combined into a single cluster. The performance of the algorithm has been demonstrated on several datasets consisting of several clusters and compared to several known clustering techniques [121].

A new grouping genetic method was introduced by Agust LE, Salcedo-Sanz S, Jiménez-Fernández S, Carro-Calvo L, Del Ser J, and Portilla-Figueras JA addressing clustering issues. This article introduces a novel grouping strategy that incorporates evolutionary operators that have undergone new adaptations as well as the idea of grouping encoding. The publication included a

detailed description of the encoding and operators as well as information on additional method implementations, including local search and parallelization with an island model. The authors tried the suggested strategy on a range of synthetic and actual clustering issues and got excellent results that outperformed established strategies like the K-means and DBSCAN algorithms [122].

A population-based worldwide search method called Particle Swarm Optimization (PSO) employs the ideas of herd social behavior. For issues with multiple peaks and complexity, PSO yields superior outcomes. In their study, Esmin, A.A., Coelho, R.A., and Matwin described earlier research on particle swarm optimization, PSO variations, and their use in clustering high-dimensional data. The research results show that PSOs used to model clusters in multidimensional space perform quite well. The rate of convergence in searches for global optimization is still insufficient, though. To address the issue of high dimensional clustering, the PSO can also be altered and successfully hybridized with other algorithms such as KFC, K-means, ACO, and GA, as well as integrated with other techniques such as dimensionality reduction and feature selection. By using these methods, high-dimensional data clustering improved, which improved prediction and data analysis in the end. Some PSO-based techniques can be used to cluster complicated, linearly inseparable datasets without knowing the exact number of clusters [123].

Researchers S. Alam, Y. S. Koh, G. Dobbie, S. U. Rehman, and P. Riddle, described the creation of a clustering technique based on particle swarm optimization. The authors presented the findings from a thorough evaluation of the growing body of research on swarm intelligence, particle swarm optimization, and PSO-based data clustering. Researchers interested in optimization-based data mining have taken notice of PSO due to the technology's innovative design and simplicity in implementation. The paper contributes in two important ways. First, the study offered a comprehensive overview of the literature with an emphasis on some of the most popular methods that have been applied to PSO-based data clustering. The analysis of the presented data is followed by a comparison of the performance of the different strategies to the most recent clustering algorithms. Additionally, it describes a PSO-based hierarchical clustering method (HPSO clustering) and evaluates the outcomes against PSO clustering, k-means, and conventional hierarchical aggregate clustering (HAC). The outcomes demonstrate that several of the most recent data clustering techniques were outperformed by PSO-based and hybrid PSO clustering strategies. The strategy is more targeted and helps prevent becoming stuck at the local optimum. Some of the most popular methods examined on benchmark datasets include hybrid PSO clustering, PSO clustering, HPSO clustering, and EPSO clustering [124, 125].

Engelbrecht, A.P and Van der Merwe, D.W., discussed using PSO on cluster data vectors in this study. There are two algorithms: the traditional gbest-PSO and the hybrid method, which tests swarms of individuals while seeding the K-means algorithm results. On six datasets, the performance of the novel PSO algorithms is assessed and contrasted with that of k-means

clustering. According to the study's findings, the PSO technique is convergent for fewer quantization errors, often greater inter-cluster distances, and generally smaller intra-cluster distances [126].

Using parallel Map Reduce techniques, Aljarah I. and Ludwig SA. suggested a scalable MRCPSO algorithm to circumvent the drawbacks of PSO clustering for large datasets. The MapReduce technique has been used to successfully parallelize the MRCPSO algorithm using commercially available hardware. In order to find the best solution for the MRCPSO clustering task based on the shortest distance between the data point and the cluster centroid, the problem is phrased as an optimization problem. The MRCPSO partitioning clustering algorithm uses centroids to represent clusters, much to the k-means approach. The velocity of the particles is used to update the centroid of each cluster. Real and artificial datasets were used in experiments to gauge the algorithm's scaling and acceleration. The findings demonstrate that MRCPSO scales very well approaches linear acceleration, and preserves acceptable clustering quality as the dataset size increases. The findings also demonstrate that, in terms of clustering quality, Map Reduce clustering outperforms the sequential K-means approach [127].

A new dynamic clustering strategy based on particle swarm optimization (PSO) and genetic algorithms (GA) was proposed by Kuo, R.J., Chen, Z.Y., Syu, Y.J., and Tien, F.C. (DCPG). The DCPG algorithm resolves the issue of determining the appropriate number of clusters depending on the features of the data and pre-specifying the number of clusters. The study proved that the DCPG method could find the right number of clusters and produce improved clustering outcomes using four benchmark datasets with various numbers, dimensions, and types of clusters. The dynamic clustering strategy based on binary-PSO (DCPSO), the dynamic clustering approach based on ACMPSO and the dynamic clustering approach based on GA (DCGA) algorithms were also contrasted with the DCPG algorithm. The PSO and GA-based DCPG algorithms exhibited quick convergence and improved clustering results. It is difficult for the solution to reach the local optimum when using the DCPG algorithm. Additionally, of all algorithms, it is the most stable. For empirical research, the Advantech Company's BOM cluster analysis was subjected to the DCPG algorithm. The suggested DCPG method identified the appropriate number of clusters, outcomes from clustering, product clusters, and common content [128].

To decrease the setup time for surface mount technology (SMT), Kuo, R. J., and Lin, L.M. suggested an evolution-based clustering algorithm based on a combination of genetic algorithms (GA) and particle swarm optimization algorithms (PSOA). For the cluster analysis in the study, the authors suggested using the two-step clustering technique ART2 + HGAPSOA. The suggested evolution-based clustering method is more accurate than the GA-based and PSOA-based clustering algorithms, according to simulation results from the Iris, Glass, Vowel, and Wine benchmark datasets. Additionally, the suggested approach outperforms GA-based and PSOA-based clustering

algorithms including GA, GKA, PSO, PSKO, GA-PSO, and GAPSKO, according to the results of model evaluations utilizing order data from multinational industrial PC manufacturers [129].

A data mining perspective on clustering methods was offered by Pham, D.T. Afify, A.A. The study focused on methods for scaling these algorithms to deal with big datasets. In order to demonstrate the potential of clustering algorithms as a tool for addressing challenging real-world situations, it also discussed several technical and engineering applications [130].

Krishnapuram, R., and Frigui, H., developed a robust clustering algorithm (RCA) based on the agglomerative hierarchical clustering method, which has applications in computer vision systems [131].

In their book, "Multi-Objective Genetic Algorithms for Clustering and Its Application to Data Mining and Informatics," authors U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay provided a thorough explanation of the use of genetic algorithms for clustering and data mining applications. Generic algorithms, multi-objective oriented optimization, soft computing, data mining, and bioinformatics were all thoroughly introduced by the authors, who then demonstrated how to systematically apply these technologies to challenges in the earth sciences, bioinformatics, and data mining [132].

Automated subspace clustering was a challenge that was overcome by Agrawal, Raghavan, Gunopulos, and Gehrke in response to the demand for new data mining applications. The suggested method, CLIQUE, is made to assist users in finding clusters embedded in high-dimensional data subspaces without assuming the subspaces may contain the clusters of interest. For easier comprehension, CLIQUE creates a cluster description in the form of a minimal DNF expression. It does not require routine data distribution and is unaffected by the input data records' order. When the number of dimensions in the data or the highest dimension in which the cluster is embedded rises, empirical examination shows that CLIQUE scales in proportion to the size of the input and has great scalability [133].

A. Dharmarajan & T. Velmurugan published a summary outlining the many fields in which partition-based clustering methods like k-Medoids, k-Means, and fuzzy c-Means are applied. The study found uses for novel and unique clustering algorithmic approaches, mainly for the medical industry [134].

Li, J. and Lewis, H.W., reviewed literature on the study of models based on fuzzy clustering in various cases and their application. In some case scenarios, modeling processes are data-driven and emphasize the distances between points and new centers of clusters. In some other cases, which aim at market segmentation or evaluation of patients by healthcare records, membership degree is a key element in the algorithm. The paper surveyed a wide range of research that has well-designed mathematic models for fuzzy clustering, some of which include genetic algorithms and neural networks [135].

Neural network-based clustering algorithms have been utilized in diabetic plantar pressure imaging datasets to obtain the key areas of foot plantar pressure characteristics [136]. The neural network-based clustering algorithms also find application in word clustering as an extension of the neural-network-based Boolean factor analysis algorithm [137]. Other areas of application include wireless sensor networks (WSNs), web users, etc [138, 139].

Density-based clustering algorithms find application in clustering spatial databases with noise, large databases, real-time stream data, mining biomedical images, time-series gene expression in bioinformatics, etc [86, 140–143].

Kernel-based clustering algorithms are used for outlier detection and market segmentation, MRI image segmentation, online clustering, and MRI brain image segmentation [144–147].

Spectral clustering algorithms have a lot of applications due to their robustness in real-world data. Some of the applications of spectral clustering algorithms are resource scheduling for edge computing, encoding constraints in constraints in algorithmic settings, medical image analysis, summarizing similarity of gene activity across different tissue samples, superpixel segmentation, segmentation of 3D meshes, automatic video analysis, single-cell RNA sequencing data analysis, biological sequence data, partitioning higher-order network structures, etc [148–151], [152, p. 3], [153–157].

A thorough analysis of particle swarm optimization techniques and their applications was done by Zhang, Y., Ji, G., and Wang, S. The authors presented research on PSO applications in eight disciplines, including mechanical engineering, fuel and energy, automated control systems, communications theory, medicine, chemistry, and biology [158].

2.2. Text Clustering

The practice of collecting interesting and important content from unstructured text is known as text mining, also known as text data mining or text knowledge discovery (KDT). Data mining, machine learning, statistics, computer linguistics, and information retrieval are all used in the emerging, interdisciplinary discipline of text mining. Since the majority of data (over 80%) is kept as text, text mining is thought to have a large commercial potential. There are many places where knowledge may be found, yet the unstructured text is still the easiest way to get knowledge [159]. Text mining is the practice of extracting relevant data and knowledge from unstructured text. Hotho, A., Paaß, G., and Nürnberger referred to text mining as a nascent, multidisciplinary field that crosses information retrieval, machine learning, statistics, computational linguistics, and particularly related data mining fields. For preprocessing, classification, clustering, information extraction, and visualization, the authors identified fundamental analytical tasks. In the research paper, numerous effective applications of text mining were briefly discussed [160].

Today, all paper documents are provided in electronic format due to their fast access and low storage capacity. Therefore, getting the relevant documents from a larger database is a big problem. Text mining is not a stand-alone task normally handled by human analysts. The goal is to convert text that consists of everyday language into a structured database format. In this way, heterogeneous documents are summarized and displayed in a unified way.

Document clustering was defined by Sathiyakumari K, Preamsudha V, Manimekalai G, and Scholar MP as the division of a set of texts into subgroups based on content similarity. The goal of document clustering, according to the study, is to satisfy people's needs for information retrieval and comprehension. The authors also offered an overview of several text clustering methodologies and algorithms in document clustering, providing interested readers with a thorough overview of the available methods [161].

It was demonstrated empirically in this study by Liu T, Chen Z, Liu S, and Ma WY., that feature selection strategies can enhance the effectiveness and performance of text clustering algorithms. The authors also ran comparison research on several feature selection approaches for text clustering, including Entropy-based (En), Document Frequency (DF), Information Gain (IG), Term Strength (TS), and statistical (CHI). They also presented a novel feature selection method named "Term Contribution (TC)". They also put out a method called "Iterative Feature Selection (IF)" that handles label shortage issues by iteratively choosing features and performing clustering. This method addresses label shortage issues. Extensive experimental findings using Web Directory data demonstrated the suggested algorithm's superior performance [162].

A number of unsupervised feature selection strategies, including Term Contribution - TC, Document Frequency – DF, Term variance quality - TVQ, and a recently suggested method Term Variance - TC, were examined by Liu, Yu, Kang, and Wang (TV). According to the study, TC and TV are superior to DF and TVQ. The study also revealed that different cluster validity criteria perform differently, with Folkes and Mallows index – FM and averaged accuracy - AA criteria outperforming others in terms of assessing the clustering results [163].

To solve the feature selection problem, Khader, A.T., and Abualigah, L.M., proposed combining particle swarm optimization methods and gene operators. K-means clustering is used to assess how well the recovered feature subsets perform. Eight well-known text datasets with various feature sets were used in the experiments. The outcomes demonstrated that by producing a new subset of more beneficial features, the suggested Hybrid Feature Selection Method (HFSPSOTC) enhanced the performance of the clustering algorithm. Other algorithms that have been published in the literature were contrasted with the suggested approach. The research found that the suggested feature selection method aids in the correct clustering of algorithms [164].

Three feature selection techniques with feature weighting schemes and dynamic dimensionality reduction were proposed by Abualigah, L.M., Al-Betar, M.A., Khader, A.T., and

Alomari, O.A. for challenges involving text document clustering. Using suitable merit functions that depend on the frequency of phrases, text documents are often sorted into numerous interconnected clusters based on carefully chosen and informative qualities. The feature selection method is used to choose the information features for each document. The most effective feature selection techniques created utilizing a novel weighting scheme, Length Feature Weighting (LFW) determined by word frequency, are the Harmonic Search Algorithm (HS), Genetic Algorithm (GA), and Particle Swarm Optimization Algorithm (PSO). The characteristics of other documents have an impact on the look. Additionally, a brand-new Dynamic Dimensionality Reduction Technique (DDR) is offered to cut down on the number of features used in clustering and boost the algorithm's speed. Finally, we aggregate groups of text documents based on terms (or features) discovered during dynamic reduction using the popular clustering technique k-mean. Seven benchmark text datasets for text mining are analyzed, ranging in size and complexity. Particle swarm optimization with length feature weighting and dynamic reduction offers the best results for practically every dataset studied, according to analysis using k-mean. The text mining community is given a new option in this paper for clustering text documents with pertinent practical qualities [165].

The popular techniques for extracting text features were defined by Liang, H., Sun, Y., Sun, X., and Gao. They then extended their research to widely used deep learning techniques in text feature extraction and its applications, and they anticipated the usage of deep learning in feature extraction [166].

Clustering the document to pre-process the text is another method for obtaining information from the text. The success of knowledge extraction is significantly influenced by the preprocessing processes. The dimensions of the text were preprocessed and reduced using TF-IDF and Singular Value Decomposition (SVD) dimensionality reduction techniques by Kadhim, A.I., Ahamed, N.H., and Cheah, Y.N. The suggested system is an efficient dimensionality reduction and pre-processing method that facilitates document clustering with the k-means algorithm. Additionally, utilizing simulated results from the BBC News and BBC Sports datasets, experimental results demonstrate that the suggested strategy enhances the performance of clustering of English text documents [167].

In this study, Shepherd, M., Tang, B., Heywood, M.I., and Milios, E., used three benchmark datasets from different genres to compare the performance of six dimensionality reduction methods when applied to text clustering problems. Based on all the results, we found that: For feature conversion methods, independent component analysis (ICA) can be rated as the best, then latent semantic indexing (LSI), and finally random projection (RP) for clustering accuracy and stability. Both ICA and LSI deliver the best performance in dimensions, most often less than 100, and in some cases less than 10. ICA and LSI maintain the best performance over a wide range of dimensions. ICA seems to be more stable than LSI. For feature selection methods, mean Tf-Idf (TI) and term frequency variance (TfV) are superior to document frequency (DF). The best results from

TI and TfV can be the same as those from ICA and LSI but at a much higher dimension. The results of combining ICA with TI or TfV thresholds are the most interesting [168].

The issue of document or textual content clustering has been addressed using a variety of methodologies. In their study, Nikhath, A.K., and Subrahmanyam, K., conducted a thorough analysis of various clustering methods for text-based data mining. The known clustering models that have been suggested were also included in the white paper, along with each model's advantages and disadvantages. Additionally, the difficulties of clustering in the text domain as well as the current state of several clustering tasks in social networks and comparable contexts were explored [169].

In this work, Shah, N. and Mahajan, S., gave a detailed review of more than thirty text document-clustering papers that used the semantic approach. The authors provided a measure of semantic similarity. In addition, they outlined various clustering techniques that consider semantics. All of these techniques are briefly explained and comparisons of these techniques are displayed in tabular form using various parameters such as the semantic approach used, author, the dataset used, evaluation parameters used, limits, and future work [170].

In this white paper, Hotho, A., Maedche, A., and Staab, S., demonstrated how to include background knowledge in a hierarchical format to generate various clustering aggregates from a set of documents. The authors compared their approach to a sophisticated baseline and found favorable results for the approach. In addition, the study showed that different views of the same input could be automatically generated. This allows the user to rely on the hierarchy to control and possibly interpret the results of clustering. The Proposed preprocessing method, COSA is a very common method of preprocessing input data by applying ontology-based heuristics for feature aggregation and selection. Therefore, it is possible to build some alternative text representations [171].

Hotho, A., Stumme, G., and Staab, S., in another paper showed how ontology improves text clustering by incorporating background knowledge. In the experimental assessment, the authors compared text pre-categorization from Reuters news feeds with smaller domain-based clustering techniques in Java's e-learning course. In the experiment, the improvement of the result by the background knowledge compared with the baseline without the background knowledge was displayed in many interesting combinations [172].

Biemann, C., in his paper, surveyed methods for ontology learning. The paper gave an in-depth explanation of ontology and its usage. The study discussed various methods to apply, construct and extend ontologies using unstructured data sources. The study focused on papers that dealt with ontology learning as well as other papers that have similar goals but with different terminology. The authors also conveyed various methods of evaluating ontologies [173].

Ontology Learning and Population from Text: Algorithms, Evaluations, and Applications were written by P. Cimiano. For academics working on text mining, information retrieval, natural language processing, the Semantic Web, etc., the book proposed a technique for ontology learning from text and its applicability. The book, on the one hand, contains introductory material and much more related work, and on the other hand, it contains detailed explanations of algorithms, evaluation methods, and more. It also provided a comparison of different ontology methods to guide engineers as well as an analysis of the impact of ontology learning in related application areas [174].

In this research, Punitha S.C. and Punithavalli M. investigated the operation of two cutting-edge methods for clustering text content. The first strategy is a hybrid strategy that combines a procedure for pattern identification with a semantic-driven method (HSTC) for grouping documents. The model referred to as HSTC (Hybrid Scheme of Text Clustering) used semantics-based distance metrics to calculate the similarity between documents by performing content and behavioral analysis. This had the advantage that lexical and structural characteristics were taken into account in addition to the stylistic characteristics of the processed document. The authors used a radial basis function (RBF) for clustering. The second technique performs feature selection text document clustering document (Text Clustering with Feature Selection (TCFS)) using an ontology-based approach. The TCFS system is designed to identify semantic relationships using an ontology that describes the relationships between terms and concepts. Conceptual weights are calculated from these relationships and used during clustering. The performance of both methods selected is evaluated in terms of clustering efficiency and clustering speed. Experiments showed that both techniques are efficient in the clustering process, but TCFS performance was slightly better in terms of clustering quality, but slower [175].

In their study, Beil, Ester, and Xu developed a novel method for grouping text documents using frequent itemsets (term sets). Using association rule mining techniques, which evaluate the overlap of frequent sets with respect to sentences in the supporting document to categorize them based on phrases that contain frequent terms, such frequent sets can be effectively discovered. The paper developed two algorithms: HFTC for hierarchical clustering and FTC for flat clustering for frequent term-based text clustering. Both traditional text documents and web documents were subjected to experimental assessments, which demonstrated that the suggested methodologies accomplish comparable quality clustering significantly more quickly than existing text clustering algorithms. Additionally, the suggested approach offers a clear explanation of the discovered clusters using frequent sets [176].

According to Li, Y., Holt, J.D., Chung, and S.M. Clustering based on Frequent Word Sequences (CFWS) and clustering based on Frequent Word Meaning Sequences (CFWMS) are two new text-clustering techniques they proposed. The meaning of a word was defined by the writers as an idea represented in the form of a synonymous term, and a word was defined as the form of a

word shown in a document. When a word (meaning) sequence appears in more than a predetermined number of documents in a text database, it is said to be common. Regular word (meaning) patterns can offer concise and useful details about these text documents. The CISI document of the Classic dataset, the corpus of the Text Retrieval Conference (TREC) datasets and the Reuters 21,578 text collection were all used in the study's tests. The experimental results demonstrated that CFWMS and CFWS have much higher clustering accuracy than the Frequent Item set-based Hierarchical Clustering (FIHC), Bisecting k-means (BKM), and modified Bisecting k-means utilizing background knowledge (BBK) algorithms [177].

The use of neighbors and connections to obtain global information to gauge the proximity of two text documents was suggested by Luo, C., Chung, S.M., and Li, Y. The algorithm is based on the premise that two documents are neighbors if they are sufficiently similar to one another, and the connection or link between two documents is the number of their shared neighbors. The k-means algorithm family has used the neighbor-and-link strategy in three ways. a new similarity employing a fusion of cosine and link functions, a new heuristic function for picking clusters to be split based on the vicinity of the cluster centroid, and a new method to choose the initial cluster centroid based on the rank of the candidate document. The proposed strategy might greatly increase document cluster performance in terms of accuracy, according to experimental results on real-world datasets, without significantly increasing execution time [178].

Mehta, V., Bawa, S., and Singh, J., proposed a clustering technique that is particularly well suited for large text datasets and overcomes the limitations associated with large text datasets, which are sparse and high dimensionality. The proposed technique is based on word embeddings derived from a recently published deep learning model called "Transformer-based bidirectional encoder representation". The proposed method is called WEClustering, and the algorithm combines the benefits of a statistical scoring mechanism like TF-IDF with a state-of-the-art deep-learning model like BERT. WEClustering first uses the BERT model to extract the embeddings of all words in a document and then combines them to form a word cluster with similar meanings and contexts. The proposed method effectively addresses the high dimensional problems and forms more clusters that are accurate. The technique is validated on several datasets of varying sizes and its performance is compared with other widely used and state-of-the-art clustering techniques. The experimental comparison demonstrates that the proposed clustering technique gives a significant improvement over other techniques as measured by metrics such as Purity and Adjusted Rand Index [179].

In their study, Kamel, M., Shehata, S., and Karray, F. bridged the gap between text mining and natural language processing. A brand-new, four-part concept-based mining methodology was suggested in order to enhance the quality of text clustering. The study showed that by utilizing the semantic structure of the sentences in the document, improved text clustering results can be obtained. The first component is sentence-based conceptual analysis. It uses the proposed

conceptual term frequency CTF measure to analyze the semantic structure of each sentence and capture the concept of the sentence. The second component, document-based conceptual analysis, then uses the concept-based term frequency TF to analyze each concept at the document level. The third component uses the global document frequency measurement DF to analyze corpus-level concepts. The fourth component is a concept-based similarity measurement. This allows the measurement of the importance of each concept related to sentence semantics, the topics in any document, and the distinction between documents in a corpus [180].

In this article, Hotho, A. and, Stumme, G., explained how to combine the efficiency of common (non-conceptual) clustering techniques with the deliberate explanation provided by the conceptual clustering approach. The study demonstrated how this method might be used with a corpus of documents. First, the documents used BiSect–k–Means clustering algorithm, with a thesaurus as background knowledge. This clustering reduces the number of documents and allows formal concept analysis to be conceptually clustered in the second step [181].

A study project on text document clustering based on optimization techniques was described by Jiji, D.G.W., and Jensi, R. In the first section of this study, data clustering is briefly introduced. a review of numerous studies on text document clustering, soft computing, and data mining. Vector space models, particle swarm optimization, bee algorithms, genetic algorithms, latent semantic indexes, ant colony optimization, and other methods are among the many methodologies discussed [182].

Liu, F. and Xiong, L., introduced common text clustering algorithms, analyzed and compared a few components of clustering algorithms which includes the relevant scope, the preliminary parameters, termination situations, and noise sensitivity. Algorithms studied include hierarchical clustering, partitioned clustering, and density-based and self-organizing maps algorithms [183].

C.C. Aggarwal & C. Zhai gave a thorough analysis of text clustering. They discussed the principal text clustering algorithms, including feature selection and transformation for text clustering, word- and phrase-based clustering, probabilistic document clustering and topic models, distance-based clustering algorithms, text network clustering, online text stream clustering, and semi-supervised clustering, as well as their relative merits. They also talked about the key developments in text clustering research as they applied to text data from dynamic and diverse applications [184].

Larsen, B., and Aone, C., described a set of algorithms, including a text clustering system that includes scatter/gather (buckshot, fractionation, split/join) and k-means techniques, and how to evaluate their effectiveness. The authors used F-Measure (a combination of precision and recall) to measure the quality of the generated hierarchy and averaged the results in ten trials. The results support two feature extraction techniques. First, weighting the extracted terms with tf-idf gives better results than using just the term frequency for all corpora except the corpus with the fewest

documents. Second, truncation of feature vectors saves time at the expense of cluster quality (clustering time and vector length seem to be very roughly proportional). The experiments suggest that continuous center adjustment contributes to cluster quality over seed selection [185].

Pantel, P., and Lin, D., presented Clustering by Committee (CBC), a clustering algorithm that automatically recognizes word meanings from the text. First, the algorithm finds well-distributed tight clusters called committees and uses them to build the centroids of the final cluster. The algorithm then continues by matching the words to the most similar clusters. After assigning an item to a cluster, duplicate features are removed from the item, allowing the CBC to discover less frequent senses of words and avoid double meanings. Each cluster to which a word belongs represents one of its meanings. The authors also presented an evaluation method for automatically measuring the precision and recall of detected senses. Experiments conducted by the authors showed that CBC is superior to some well-known hierarchical, partition, and hybrid clustering algorithms [186].

An algorithm for co-clustering documents and words called Fuzzy CoDoK was announced by Kummamuru, Krishnapuram, and Dhawale. The Fuzzy Clustering for Categorical Multivariate Data (FCCM) technique has been modified to create Fuzzy CoDoK. The suggested technique, in contrast to FCCM, can handle document clustering when there are many documents or words. The authors' experiments have demonstrated that the updated approach is scalable and generates useful clusters. The study also introduced the Spherical Fuzzy c-Means (SFCM) algorithm and showed how the FCCM and Spherical K-Means (SKM) algorithm interact [187].

The Core Phrase algorithm was developed by Hammouda, K.M., Camel, M.S., and Matute, D.N. for precisely extracting informative key phrases from textual clusters. In comparison to keyword-based cluster tagging methods, the approach is domain neutral and offers high accuracy in determining the subject of a document cluster. The approach can also be used to precisely identify clusters and group clusters for other reasons, such as figuring out how similar clusters are to one another or how similar articles are to one another [188].

Wang, A., Li, Y., and Wang, W., proposed a KP-FCM clustering method that uses key phrases as text features and applies fuzzy c-means (FCM) as a clustering algorithm. The algorithm worked well on both English and Chinese datasets in terms of overall accuracy, overall recall, and overall F-Measure. Since this method uses bag of "key phrases", the method can also give each cluster a readable phrase label (brief description) [189].

In order to process relational input data, Skabar, A., and Abdalgader, K., introduced a new fuzzy clustering algorithm. Specifically, information that represents the pairwise similarity between data objects as a square matrix. This algorithm utilizes an Expectation-Maximization framework with a graph visualization of the data. The framework evaluates the probability of an object on the chart based on its network centrality. The algorithm can discover overlapping clusters of

semantically related sentences, which can be beneficial in a variety of text-mining applications, according to the results of applying it to a sentence-clustering task [190].

Introducing a brand-new information- based vector space model (VSM) for text document clustering are Jing, L., Huang, J.Z., and Ng, M.K. The representation of a written document as a collection of vectors in the new model incorporates semantic links among terms (concepts, words, etc.). The goal is to enhance text clustering outcomes by more accurately calculating the dissimilarity between two documents. The degree to which two concepts differ from one another determines their semantic relationship. The term frequency of VSM is reweighted using these analogies. To determine the semantic link between two terms based on two separate techniques, the authors looked at and investigated two different similarity measures. The first method uses existing ontologies like WordNet and MeSH and combines edge counting, mean distances, and position weighting methods to determine how similar two terms in an ontology hierarchy are to one another. The second method built relationships between terms and determined their semantic similarity using a text corpus. The actual textual data represented by the new knowledge base VSM was grouped using three clustering techniques: feature weight k-means, bisecting k-means, and hierarchical clustering algorithms. According to experimental findings, the performance of clusters using the new model is significantly superior to that of clusters using conventional concept-based VSMs [191].

Zhong, S., and Ghosh, J., presented detailed twelve generative approaches to text clustering achieved by applying four document-to-cluster allocation strategies (hard, probabilistic, soft, and deterministic annealing (DA) base assignment) to each of the three base models: the multivariate Bernoulli, the multinomial distribution, and the Von Mises Fisher - (vMF) distribution. Empirical results on a large number of high-dimensional text datasets highlighted the following trends: The Bernoulli model is not suitable for text clustering; The von Mises Fisher model is often superior to the multinomial model for clustering documents; Algorithms that use soft assignments slow down the execution of Bernoulli and multinomial models and slightly improve cluster performance [192].

Zhao, Y., and Karypis, G., concentrated on comparing the clustering outcomes of each criteria function of agglomerative clustering algorithms with partitioning clustering algorithms and experimentally evaluating the performance of seven different global criteria functions in the context of agglomerative clustering algorithms. Experimental evaluation showed that the partitioning algorithm always yields better clustering results than the agglomerative algorithm in regards to any criteria function. This suggests that the partitioning clustering algorithm is suitable for clustering large datasets due to its relatively small computational effort, and better cluster performance [193].

Xu, J., Huang, J.Z., Jing, L., and Ng, M.K. provided a novel approach to the clustering of big and complicated text data. The approach is based on a fresh subspace clustering algorithm that computes feature weights for the k-means clustering procedure automatically. The detection of

clusters in subspaces of the document's vector space and the identification of keywords that describe the semantics of the clusters are done using feature weights when clustering sparse text data. The authors provided a solution to the sparsity issue that arises with text clustering by modifying the FW-k-means algorithm. The new method outperforms the bisection k-means and regular k-means algorithms while maintaining the effectiveness of the k-means clustering process, according to experimental results utilizing real-world text data [194].

In this research, Steinbach, Kumar, and Karypis (Steinbach, Kumar, and Karypis) presented the findings of an experimental investigation of three widely used document clustering methods. Agglomerative hierarchical clustering and k-means were the two basic methods for document clustering that were specifically compared in the study. The authors employ both the "conventional" K-means method and a K-means variant known as "bisecting" K-means (for k-means). The study's findings show that the bisecting K-means strategy performs as well as or better than the evaluated hierarchical approaches and is superior to the regular K-means approach. More specifically, a measure of cluster quality entropy and total similarity shows that the bisected k-means strategy yields a very reliable and excellent clustering solution. In addition, when building a document hierarchy (measured with an F measure), bisecting k-means seems to be slightly better than UPGMA, the best hierarchy technique [195].

The majority of conventional clustering methods fall short of the demands of document clustering, which include large volumes, high dimensionality, and easy browsing with informative cluster labels. Fung, Ke Wang, Benjamin CM., and Martin Ester presented a novel strategy for tackling these problems in this study. This method is unique in that it defines clusters, arranges cluster hierarchies, and minimizes the dimensions of document sets using frequent itemsets. The accuracy, effectiveness, and scalability of the suggested approach were demonstrated through experiments to be superior to those of conventional clustering algorithms [196].

To find a hierarchical clustering solution for document datasets, Zhao, Y., Karypis, G., and Fayyad experimentally examined nine agglomerative methods and six partitioning strategies. Compared to the agglomerative method, the partitioning method yields a more superior hierarchical solution, according to the experimental findings. Combined with previous work on the effectiveness of various partitioning algorithms for building k-means clustering solutions (Zhao and Karypis, 2004), the authors found that partitioning methods are suitable for producing both flat and hierarchical clustering solutions for document datasets. They also introduced a new class of agglomerative algorithms by constraining the agglomerative process using clusters obtained by the partition algorithm. Experimental results have shown that constrained agglomeration improves the clustering solution obtained as opposed to results obtained by the agglomerative or partitioning method alone [197].

Pons Porrata and Gil Garca describe two clustering techniques in this work under the names Dynamic Hierarchy Compact and Dynamic Hierarchy Star. The goal of both approaches is to create a cluster structure that can manage dynamic datasets. While the second maintain overlapping hierarchies, the first creates disjointed hierarchies of the cluster. These methods are effective and efficient for developing hierarchical clustering solutions in dynamic situations, according to experimental results from multiple benchmark text datasets, but they are also simpler to search than conventional algorithms. means to provide. Consequently, it is advised for activities that call for dynamic clusterings, such as information organization, the creation of document taxonomies, and the identification of hierarchical subjects [198].

For text document datasets, Sahoo, N., Krishnan, R., Callan, J., Padman, R., and Duncan, G. examined incremental hierarchical clustering algorithms that are frequently employed for non-text datasets. They also suggested an alternative with improved characteristics for incremental hierarchical text clustering. In place of the normal distribution used in the Classit algorithm's initial formulation, the Cobweb / Classit algorithm version demonstrated in this task employs the Katz distribution. It has been demonstrated in the literature and empirically seen in the suggested model that the Katz distribution is better suitable for word occurrence data. The Reuters RCV1 dataset was used by the authors to compare the two algorithms. As a result, the authors may conduct experiments in settings that are remarkably comparable to real-world ones. The outcomes reveal that the suggested strategy yields superior quality clusters than the conventional Cobweb / Classit algorithm [199].

C.C. Aggarwal, and P.S. Yu, described how to cluster data streams of text and categories using a compact summary representation of clustering statistics. The proposed algorithm can be used for both text and categorical data mining domains with minor changes to the underlying summary statistic. Experimental tests showed that the algorithm adapts quickly to temporal fluctuations in the data stream [200].

The semantic smoothing model was expanded in this article by Liu, Y.B., Cai, J.R., Yin, J., and Fu, A.W.C. in the context of text-based data streams. The authors provided two online clustering methods, OCTSM, and OCTS, for clustering big text data streams based on the expanded model. In this study, a novel cluster statistical structure dubbed the cluster profile was introduced by both algorithms. This allows dynamic capturing of the semantics of a text data stream while accelerating the clustering process. Some efficient implementations of the algorithm are also shown. Finally, the researchers published a series of experimental results explaining the effectiveness of their technology [201].

For quick and adaptable clustering of text streams, this white paper by S. Zhong combines an effective online spherical k-means method (OSKM) with currently available scalable clustering strategies. The OSKM algorithm uses the well-known Winner-take-all competitive learning-based

online update to modify the spherical k-means (SPKM) algorithm (for cluster centroids). The algorithm has been shown to be significantly more effective than SPKM and produces significantly higher-quality clustering. The purpose of this method is to handle limited storage by keeping (part of) the historical data contribution. The authors included a forgetting factor that applies exponential decay to the significance of historical data in order to modify the suggested clustering method to the data stream. The weight of a set of text documents decreases with age. The effectiveness of the suggested approach was demonstrated by experimental findings, which also revealed several logical and fascinating details about text stream clustering [202].

One of the most popular text analysis techniques for obtaining knowledge from social media websites like Facebook, Twitter, Instagram, and Weibo is short text clustering. A topic representative term discovery (TRTD) technique for short-text clustering was put out by Yang, S., Huang, G., and Cai in their publication. TRTD finds groupings of significant terms that are strongly related to one another as a group of terms that indicate a topic and may previously detect noisy and insufficient topic terms using node-weighted and edge-weighted word graphs. The concerns of short text clustering's sparsity and noise are likewise addressed by TRTD. The proposed strategy outperforms seven alternative methods in terms of accuracy, efficacy, and efficiency, according to extensive testing on real-world datasets [203].

This paper introduces AntSA, a novel AntTree-based technique for clustering short text corpora, by M.L. Errecalde, D.A. Ingaramo, and P. Rosso. At different points during the clustering process, AntSA incorporates knowledge about silhouette coefficients and the idea of attraction. AntSA is a general technique that enables the use of several clustering algorithms to obtain the initial data division and definition of formulae for different attractions [204].

The authors of this study, C. T. Zheng, H. San Wong, and C. Liu, developed a text extension method based on the short text data itself. Corpora-based extension is what it is. The authors developed a virtual topic ratio vector from the words in a given batch of texts by introducing the word-topic conjugate definition. Virtual words are then formed from these vectors. By doing this, the short text can be improved with the possible word's posterior probability if it appears in the original document. The virtual document utilized for enrichment is semantically consistent with the original material, which is one of the properties of the suggested approach. The corpus-based extended technique considerably enhances the performance of short text clustering results, according to experiments with various datasets [205].

A novel clustering method called TermCut was proposed by Ni, X., Lu, Z., Quan, X., Hua, B., and Wenyin, L. for clustering brief passages of text. This technique's goal is to compile short text samples into groups based on key terms and commonalities among them. Specifically, the proposed reduction of the RMcut criteria is used to identify the core terms of each cluster. Short text snippets have been proposed to be clustered using the TermCut method using two distinct

clustering algorithms, Threshold-based TermCut (TTC) and Cluster Number-based TermCut (CNTC). The employment of the two algorithms for various purposes makes them different from one another. When the appropriate number of final clusters can be acquired, CNTC is applied. But if the precise number cannot be identified, the TTC algorithm groups brief passages of text by designating a threshold as a stop condition. The suggested technique was tested on two distinct datasets, and the outcomes demonstrated that it was successful at clustering short text snippets, including questions and search snippet data [206].

A collapsed Gibbs sampling approach for the Dirichlet-Multinomial distribution mixture-model for short text clustering was proposed by Yin, J., and Wang, J. in this study (GSDMM for short). The Movie Group Process (MGP) was also suggested by the authors as a GSDMM comparison. The reader will gain a better understanding of GSDMM's operation, motivation, and significance of its parameters as a result. GSDMM is capable of automatically generating a variety of clusters that balance the consistency and integrity of clustering outcomes and can converge quickly. The sparse and high dimensional problem of short texts may also be handled by GSDMM, and the representative words for each cluster can be found. A thorough experimental research has shown that GSDMM can perform noticeably better than the standard approaches [207].

Song, W., & Park, S.C., proposed a genetic algorithmic technique for text clustering based on the Latent Semantic Index Model (GAL). GAL determines the ideal amount of clusters to give text clustering that is close to optimal automatically. Analysis reveals that LSI not only offers the semantic foundation for the text model but also drastically shrinks the dimensions. This is a great scenario for GA to create the best text clusters. The Reuter 21578 text collection is used to apply GAL as a more successful clustering algorithm than GA with VSM [208].

The particle swarm optimization text document clustering algorithm was presented by Cui, X., Palathingal, and Potok (PSO). The PSO clustering algorithm conducts a worldwide search over the solution space as opposed to the localized search of the K-means algorithm. Four separate text document datasets were used in the tests, and PSO, K-means, and hybrid PSO clustering techniques were used. There are between 204 and 800 documents in the collection, and there are between 5000 and 7000 words. The findings indicated that the hybrid PSO method can generate results for clustering that are more compact than those of the k-means technique [209].

To enhance the clustering of text documents, Janani, R., and Vijayarani, S., introduced a new spectral clustering approach called SCPSO that makes use of particle swarm optimization. The initial population is subjected to randomization while taking into account both global and local optimization functions. To handle massive text documents, the project aims to combine spectral clustering and swarm optimization. The benchmark database compares the proposed algorithm SCPSO against various current methods. The Expectation-Maximization (EM), Spherical K-means, and conventional PSO algorithms are contrasted with the suggested algorithm SCPSO. The

findings demonstrated that, in comparison to previous clustering methods, the suggested SCPSO algorithm offers improved clustering accuracy [210].

In order to enhance the clustering of web documents, Abualigah, AlBetar, M.A., L.M., Khader, A.T., and M.A. Awadallah developed two novel text-clustering methods based on the Krill-Herd (KH) algorithm in this study. The basic KH algorithm is used in the first technique along with all of its operators, whereas the basic KH algorithm's genetic operators are disregarded in the second way. Analysis and comparison of the suggested KH algorithm's performance with the k-means algorithm are conducted. In the experiment, four common benchmark text datasets were used. The findings demonstrated that the suggested KH method outperformed the k-means algorithm in terms of cluster quality, as measured by two widely used clustering indicators: entropy and purity [211].

For the purpose of clustering text documents, Abualigah, L.M., Hanandeh, E.S., and Khader, A.T., introduced the MHKHA algorithm, which combines an optimization function and a hybrid Krill Herd algorithm. In this variation, the clustering choice is based on two combined goal functions, and the k-means clustering method inherits the KH algorithm's initial solution. The Computational Intelligence Laboratory's nine standard text datasets are used to assess the effectiveness of the suggested method. Accuracy, precision, recall, F measure, and convergence behavior are the five measures used. Thirteen other methods that have been published in the literature as well as other well-known clustering algorithms are contrasted with the proposed variations of the KH algorithm. MHKHA produced the best results across all evaluated criteria and datasets assessed for clustering algorithms [212].

In the article, L.M. Abualigah, E.S. Hanandeh, and AT. Khader utilized six iterations of the krill herd algorithm to solve document clustering problems using 8 benchmark datasets (two iterations of the basic KH algorithm are adapted, three iterations of the improved KH algorithm are proposed, and an improved KH algorithm with a new hybrid function is integrated and proposed as an efficient method). Using the same datasets, the proposed krill herd algorithm is contrasted with other published algorithms such as harmonized search, genetic algorithm, k-means, and particle swarm optimization. The suggested krill herd version is evaluated according to eight criteria: precision, ASDC, recall, accuracy, F-measure, entropy, and purity. Comparing the suggested improved krill herd algorithm to previous comparison clustering approaches described in the literature, produced the best results for all benchmark datasets used [213].

Using the microblogging corpus from Twitter, Rangrej, Tendulkar, and Kulkarni investigated several clustering approaches for short text documents and produced experimental findings. The authors used Twitter short text data to examine the performance of several document clustering strategies such as graph-based, SVD-based methods, and K-means methods. There are established metrics for measuring clustering mistakes in these methods. A graph-based method

using affinity propagation is appropriate for clustering brief text data with a minimum amount of clustering errors, according to observations [214].

In their paper, Seifzadeh, Karray, Kamel, M.S., and Farahat discussed ways to increase the efficiency of clustering algorithms for short text documents. The experimental findings demonstrate that by adding information on the similarity measure derived from the statistical correlation between terms, the proposed strategy is superior to the fundamental methods. In order to produce a low-rank estimate of the Term-Term correlation matrix and alleviate the issue of a high computational time, the authors have developed a method that involves selecting a subset of terms and using the selected terms in combination with the Nystrom approximation. The terms used in the Nystrom approach are chosen at random with a probability corresponding to the length of the vector in the document space. As a result, more important terms can have a greater impact on the approximation of the term correlation matrix, and therefore higher accuracy can be achieved [215].

In this paper, Banerjee S., Gupta A., and Ramanathan K. proposed strategies to enhance expression using extra features from Wikipedia in order to increase the accuracy of short text clustering. Comparing this improved representation of text elements to more conventional representations, such as a bag of words, empirical findings reveal that it can greatly increase the accuracy of clustering [216].

In this paper, Jia, C., Wang, X., Carson, M.B., and Yu, J. developed a new concept decomposition approach, WordCom, drawing inspiration from the fact that words in the same concept frequently occur in the same short text and form a densely connected community in the word co-occurrence network. The community identification method k-rank-D is used to derive concept vectors from the word co-occurrence network because word communities can capture intricate interactions between words. This empirical investigation demonstrated that WordCom outperforms cutting-edge algorithms in terms of accuracy and that, in most instances, the top 10 words depending on prominence in each identified word community disclosed the community topic. Experiments show that WordCom can handle short texts with high dimensions and sparse data [217].

The efficiency of various distance functions and similarities, including squared Euclidean distance, cosine similarity, relative entropy, etc. utilized for clustering was compared and examined by A., Huang. The usefulness of these approaches in partition clustering of text document datasets was compared and examined in this work. Seven text document datasets are employed in the experiment, and the five most popular distance/similarity metrics for text clustering are presented along with the results [218].

Deep clustering, also known as deep learning-based clustering, was presented by Min, E., Liu, Q., Guo, X., Cui, J., Zhang, G., and Long in a thorough survey. Deep clustering classification

is suggested from the standpoint of network architecture, and sample techniques are thoroughly described. The taxonomy clearly outlines the traits, benefits, and drawbacks of the various deep clustering algorithms. They also offered a number of intriguing potential directions for deep clustering [219].

Jo, T., proposed a new representation of the document called a string vector and a new unsupervised neural network that receives a new representation as input data called Neural Text Self Organizer (NTSO) for text clustering. The study addressed two major problems in representing documents with numerical vectors for using traditional machine learning algorithms for text clustering: large dimensions and sparse distributions. Experiments have shown that string vectors represent documents more compactly than numeric vectors, eliminate large dimensional problems, and maintain robust clustering performance. Since string vectors have words as elements, sparse distributions that can occur with numbers containing zeros do not occur with string vectors. NTSO follows the conceptual method of the Kohonen network in the context of its architecture and learning rules, but the calculations related to its learning are significantly different from those of the Kohonen network [220].

Yi, J., Zhang, Y., Zhao, X., and Wan, J. proposed an approach to text clustering named deep-learning vocabulary network (DLVN). The lexical network is based on a set of related words, which contains the "co-occurrence" relationship of words or terms. They replaced the word frequency in the feature vector with the "importance" of the word in terms of vocabulary network and PageRank, which can generate a more accurate feature vector to represent the meaning of text clusters. In addition, a sparse-group deep belief network is proposed to reduce the dimensionality of feature vectors [221].

Wang, F., Tian, G., Xu, B., Zhao, J., and J. Xu, J., Hao, H., and Xu, J. presented Short Text Clustering through Convolutional Neural Networks (STCC), which is more advantageous to clustering by taking into account one constraint on acquired attributes through some kind of self-taught framework without needing any external labels or tags. They used a locality-preserving restriction while encoding compact binary codes that contained the original keyword characteristics. In order to learn deep feature representations, the word embeddings are investigated and fed into convolutional neural networks, whose output units are then trained to fit the pre-learned binary code. They employ k-means to group the learned representations after getting them [222].

Hadifar, A., Demeester, T., Sterckx, L., and Develder, C. proposed a method for clustering short texts using sentence embeddings and a multi-phase approach, starting from unsupervised smooth inverse frequency(SIF) embeddings for short texts. The short text clustering (STC) model is made up of an AutoEncoder architecture which is fine-tuned for clustering using a self-training approach that uses assignments from a clustering algorithm as supervision to update the weights of the encoder network [223].

Zhao, J., Zheng, S., Xu, B., Tian, G., and Wang, P. suggested an adaptable architecture for self-taught convolutional neural networks to cluster short texts (STC2). The framework can successfully and flexibly combine more beneficial semantic elements and unsupervisedly learn deep text representation. In the model framework, unsupervised dimensionality reduction is used to first incorporate the original raw text characteristics into compact binary codes. Convolutional neural networks are trained to learn a representation of the deep features by exploring word embeddings. During the training phase, the output units are utilized to fit the binary codes that have already been taught. Finally, they used K-means to cluster the learned representations and arrived at the best clusters [224].

Wang, B., Wei, J., Liu, C., Liu, W., Hu, X., and Lin, Z., developed a text clustering algorithm based on deep representation learning using the transfer learning domain adaptation and the parameters update during cluster iteration. First, data from the source domain is used to pre-train the deep learning clustering model. This process serves as an initialization of the model parameters. The domain discriminator is added to the model, to domain-divide the input samples. If the discriminator cannot distinguish which domain the data belongs to, the common feature space of the two domains is obtained, thereby solving the domain adaptation problem. Finally, the text feature vectors obtained by the model are grouped with the MCSKM++ algorithm [225].

Fan, Y., Zhaoying, S., Kui, M., and Gongshen, L., presented a neural feedback text-clustering algorithm based on BiLSTM-CNN-K-means architecture. Text representations are generated with contextual semantics by combining BiLSTM and CNN. A neural feedback clustering strategy using the k-means algorithm is also implemented to dynamically adjust and optimize the network training [226].

Zhang, W., Dong, C., Yin, J., and Wang, J. presented a new clustering model Attentive Representation Learning model (ARLAdv) for short text clustering, where cluster-level attention is proposed to capture the correlations between text representations and cluster representations. The learning of representation and clustering of short texts are integrated into a unified model. Additionally, by injecting adversarial disturbances into the cluster representations, adversary training is applied in an unsupervised clustering setting. The model parameters and perturbations are optimized alternately through a minimax game [227].

There have been previous attempts to discover insights into infectious diseases using machine learning. The book artificial intelligence in precision health outlined the various use of machine learning in the health industry. Specifically, Chapter 18 of the book identified potential applications of machine learning in the field of infectious diseases, which includes epidemics, antibiotic resistance, prediction of infectious diseases, and transmission of infectious diseases. The focus is on using machine-learning techniques on key aspects of infection such as diagnosis, transmission, response to treatment, and resistance. Scientists are now able to better predict

epidemics, understand the specificity of each pathogen, and identify potential targets for drug development using deep learning and machine learning models [228].

To show examples of biomedical techniques that rely on cluster analysis and to assist biomedical researchers in choosing the most appropriate clustering algorithms for their applications, Xu, R., and Wunsch, D.C., provided an overview of the state-of-the-art machine learning clustering algorithms for use in the field of biomedicine [229].

Andreopoulos, B., Wang, X., Ann, A., and M., Schroeder, presented a set of desirable clustering features used as criteria for clustering algorithms. The study reviewed forty different clustering algorithms using different approaches and data types. The authors compared the algorithms based on the desired clustering features and outlined the strengths and weaknesses of the algorithms as a basis for adapting to biomedical applications [230].

Diagnostic imaging, clinical laboratory, and mass screening data, among other types of data, have been subjected to the application of deep learning algorithms like convolution neural network (CNN), deep belief network (DBN), recurrent neural network (RNN), and various deep neural network architectures [231].

Text clustering using deep learning has also been applied in the health sector. Gencoglu, O., applied deep representation learning in the form of CNN-based AutoEncoders for clustering health tweets [232]. Karim, M.R., Beyan, O., Zappa, A., Costa, I.G., Rebholz-Schuhmann, D., Cochez, M., and Decker, S. Presented a state-of-the-art review of deep-learning-based approaches that would be useful for bioinformatics research. They also discussed in detail the training procedures of deep-learning-based clustering algorithms and also several metrics to evaluate them using bioimaging, biomedical text, and cancer genomics datasets [233]. Another study developed a new algorithm that uses a fuzzy latent semantic analysis (FLSA) Approach to discover topics in Health and Medical Corpora. The algorithm was tested on five datasets including a health news dataset collected from the Twitter account of popular news agencies [234]. Lu, Y., Liu, J., Zhang, P., Deng, S., and Li, J., suggested a way to automatically explore health-related hot topics in the online health community based on clustering analysis techniques that are better than the statistical analysis method adopted in most previous studies. By integrating medical discipline-specific knowledge, the authors built a medical topic analysis model. Using case studies, the proposed method is validated as an effective approach for automatically identifying various health-related problems. Some valuable conclusions can be drawn from the experimental results. This includes significant differences between the most commonly discussed topics in various types of disease discussion forums [235].

There have been several studies on the COVID-19 pandemic utilizing tweets from Twitter. A study on COVID-19 surveillance through Twitter using self-supervised and fewShot Learning to fine-tune a semi-supervised model built on unlabeled COVID-19 data and previously labeled

influenza was done to provide insights into COVID-19 data that have not been investigated. In this study, three datasets were used: Coronavirus (Lam-sal, 2020) for self-supervised pre-task learning using Convolution AutoEncoder, FluTrack (Lambet al., 2013), and FluVacc (Huang et al., 2017) for training the influenza classifiers. The fewShot learning was done by using the ULMfit embeddings to transfer knowledge between health and Twitter data. The ULMfit encoder parameters were frozen and only the Multi-layer perceptron (MLP) was allowed to be trained [236]. A study using deep learning classifiers for sentiment analysis was also done on Twitter COVID-19 tweets. The study used fuzzy-based rules in combination with various machine-learning classifiers to evaluate the sentiment of tweets. The study found that negative sentiments outweigh positive sentiments [237]. Lucy Lu Wang and Kyle (December 2020) also did a review of the various resources that have been introduced to support text-mining applications over the COVID-19 literature. The study specifically focused on the discussion of the COVID-19 data corpora, how the resources were modeled, systems, and shared tasks that have been introduced and invented for the COVID-19 pandemic [238]. Specifically, there have been studies made using Deep transfer learning and natural language processing for text clustering during the COVID-19 pandemic. In this study, COVID-19-related tweets were collected and analyzed in units of time (hour, day, and month) utilizing various agglomerative clustering models and pre-trained language models to understand and plot the changes in topics with time. LDA-based Topic models, sentence-BERT, and word2vec embeddings were used to model the dynamic clusters [239]. Another study was done on detecting trending COVID-19 Topics on Twitter. The study proposed a model that is based on the universal sentence encoder to detect the main topics of tweets from Twitter for a fixed number of months. The researchers first extracted the representation (embedding) of sentences in Tweets using "Sentence Transformer", which can capture the semantic information of the sentences. Thereafter, clustering algorithms were used to group similar sentences (based on their embeddings) into the same groups. Ideally, different clusters contain different semantic topics. After which text summarization is used to get the summary of sentences in each cluster, which uncovers the trending topics in the cluster [240]. A study on opinion mining using tweets from Twitter took up the challenge of mining a comprehensive set of online news texts, for determining the sentiment prevalent in the context of the ongoing pandemic. It also studied the statistical analysis of the relationship between the actual effect of COVID-19 virus infections and the online news sentiment of news agencies on Twitter. The study used AFINN, n-grams, and other statistical methods to find correlation and causation between online news tweets and statistical data about the number of cases and deaths of COVID-19 patients [241].

Sentiment analysis and topic modeling have been investigated in the coronavirus vaccine literature, however, text clustering of coronavirus vaccine tweets has not yet been done. In order to undertake sentiment analysis and topic modeling on the main worries Indian residents have

concerning the coronavirus vaccines on social media, Praveen, S.V., Ittamalla, and Deepak used machine learning approaches. Only 35% of social media posts on the coronavirus vaccine were positive, according to the study's findings, and as the incidence of coronavirus cases rises, so does public support for the vaccine [242]. In their work, Kwok, S.W.H., Wang, G, and Vadde, S.K. employed machine learning techniques to gather themes and sentiments on Twitter about coronavirus immunization in Australia. According to the findings, the three LDA subjects were misconceptions and grievances over coronavirus management, advocacy of infection control measures against coronavirus, and attitudes toward coronavirus and its immunization. Almost two-thirds of all tweets reflected favorable public sentiment toward the coronavirus vaccination. About one-third was negative [243].

3. RESEARCH METHODOLOGY

This chapter presents the research methodology and theories that are necessary to perform text clustering and topic modeling of coronavirus vaccine tweets.

3.1. Introduction

Large amounts of information are now accessible to everyone thanks to advances in information and communication technology, which have also caused an exponential rise in the volume of text that is available online. As increasing amounts of documentation are made available electronically, efficient retrieval and extraction are becoming increasingly challenging without summarization, effective document structure, and indexing. Different methods have been suggested to address the issue, with text document clustering, a branch of text mining, being the main one [244].

Text mining can be defined as a process of solving problems specifically in areas such as Information retrieval from documents, information extraction, text classification, text clustering, etc., and leveraging methods from related fields such as natural language processing and knowledge base (KB) construction. Automated document organization, information retrieval, topic extraction, and filtering all have one thing in common. They require text clustering (sometimes also known as document clustering) to be done quickly and accurately.

Text clustering is the application of cluster analysis to textual documents. Natural language processing (NLP) and machine learning are used to comprehend and classify unstructured text-based data [238]. The text clustering problem can be described as follows: Given a collection of N documents, denoted as D_N and a predefined cluster amount K (typically assigned by the user), D_N is divided into K text document clusters D_{n1}; D_{n2};....., D_{nk}, {i.e.; D_{n1}; D_{n2};....., D_{nk}} = D_N so that documents within a cluster are similar to one another and the documents within differing clusters are different [245]. This is shown in figure 3.1 below.

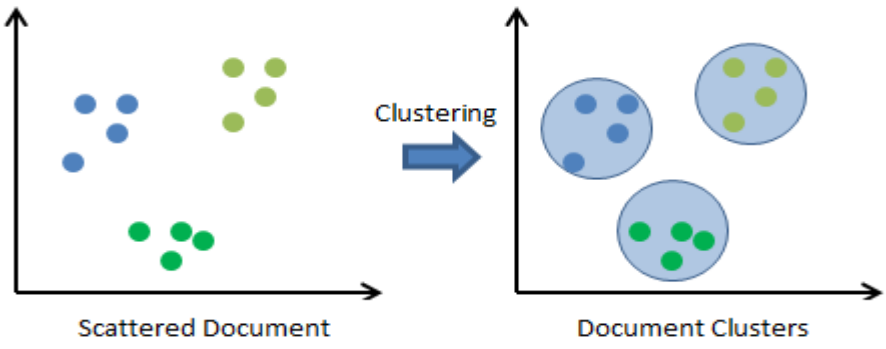


Figure 3.1. Text clustering of scattered documents.

3.2. Text Representation

Contrary to the conventional clustering task, text document clustering has additional difficulties. While documents are sparse, varied in length, and can contain associated phrases, text corpora are high dimensional in terms of word count. In most cases, the structure of the text information is limited or even unstructured, because the document contains natural human language, so the problem in text document clustering is how to structure the representation of the text to facilitate mathematical processing and analysis. The clustering task's success depends on identifying a document representation model—a set of attributes that can distinguish between documents. Words are features in the text representation model that describe the object, which is the text. The chosen document representation model has a significant impact on both the clustering technique and the metrics used to determine document similarity between documents [1, 184].

3.2.1. Probabilistic Models

For the document clustering problem, given a corpus of texts, or a collection of documents, $D = \{d_1,.....,d_N\}$, called a corpus. The vocabulary of D is represented by the set of words $V = \{w_1, \ldots\ldots, w_M\}$. Each sentence in the document $d \in D$ is a sequence of n_d words. Where d represents a document's word vector. occasionally, d can be described as consisting of contiguous, non-overlapping textual units, or segments, which themselves are made up of sentences and words. A segment-set, S, is a collection of segments. The set of segment sets from a particular document d is denoted by S_d while the set of segment sets from all the documents in a set, D is denoted by $S = U_{d \in D}{}^{Sd}$. A document clustering yields the set $C = \{C_1, \ldots\ldots, C_K\}$ of clusters.

In order to connect hidden subjects - unobserved class attributes with words in a document - observed data, probabilistic generative algorithms train a low-dimensional feature space model. For this group of algorithms, the following notations are appropriate: The documents are shown as word sequences (instead of word sets), with $di = (w_1, \ldots\ldots, w_{ndi})$. The text document's words are depicted as unit-basis vectors, wherein w_l is a vector of size M with $w_l{}^u = 1$ and $w_l{}^v = 0$ for all indexes $u \neq v$, and its segments are also likewise considered as word sequences if the document is divided into sections. But a document can also be a series of parts or a set of sequences. Each subject in the set of latent feature topics z_k in $Z = \{z_1, \ldots\ldots, z_K\}$, is a distribution over the vocabulary D. The proportion of the document or segment that can be pulled from each subject is specified by topic proportions, e.g., α and θ. The z and y topic assignments, respectively, specify which subject is chosen as the source of the chosen phrase or document. To illustrate graphically the complexity of certain models, plate notations are commonly used to express probabilistic generative models. In this notation, the rectangles (plates) stand in for repeated sections of the model. The number in the plate's lower right corner denotes how many times the included variables were repeated.

Unobserved and observed latent variables are represented by un-shaded and colored variables, respectively. The models in figure 3.2 (a) and (b), where *M* words are chosen at random from a distribution β, are identical. Plate notation makes the description in (b) more condensed.

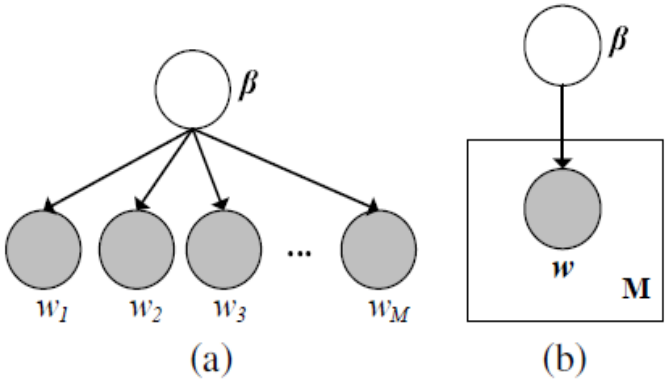


Figure 3.2. Examples of graphical model notations For the same model shown in (a), the plate notation in (b) offers a more condensed notation of (a).

When a text is represented as a collection of topics by the probabilistic topic model, subjects are the probability distribution of the words. The issues with the term as a topic are solved by this model. A term can refer to a word or a phrase. Complex subjects, vocabulary changes, and inconsistencies in word meanings cannot be represented by it. The two most well-known topic-modeling techniques are Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Indexing (PLSI).

3.2.2. Vector Space Models

A word (or feature, more generally) in each document is represented as an element of a vector in a very high-dimensional space using the bag-of-words representation that has been used traditionally to describe document collections. This representation is based on a concept of vector space in which the vector components correspond to certain feature weights. In general, the vector space model is used to describe a document D as a vector of term weights $d =< w_1, w_2, \dots \dots w_{|V|} >$, where V is the collection of words (also referred to as features) that appear at least once in the set of document D_N. In an n-dimensional space, wherein n is the number of distinct terms in the lexicon, documents are presented as vectors. Different weighted schemes, including Term Frequency (TF), Boolean value, Inverse Document Frequency (TF*IDF), Inverse Document Frequency (IDF), and Term Frequency, can be used to calculate each term's weight value. Document representation using the vector space model does not take the semantic relationships between terms into account. In other words, it is presumptive that the concepts represented in the vector space are independent of one another. It is popular because it is straightforward, helpful for expressing document properties, and very simple to calculate how similar two texts are.

3.2.3. Statistical Models

An example of a probability distribution over word sequences is a statistical language model. Statistical language models are necessary for a variety of language technology applications. Information retrieval, optical character recognition, document categorization and routing, speech recognition, spelling checkers, handwriting recognition, and many more processes fall under this category. The n-gram language model is one of the most effective and well-liked statistical language modeling methods. The n-gram language model makes the assumption that a word's likelihood relies on the n words that came before it. In other words, the probability $P(w_1, w_2, w_3, \ldots\ldots w_m)$ of observing the sentence $w_1, w_2, \ldots\ldots\ldots w_m$ is approximated as:

$$P(w_1, w_2, \ldots\ldots w_m) = \sum_{i=1}^{m}(w_i|w_1, \ldots\ldots\ldots\ldots w_{i\_1)}. \quad (3.1)$$

3.2.4. Generative Models

The generative models provide grouping techniques for texts that have been statistically classified by subjects and explain a statistical model of such materials. The main concept behind this study is that the bag-of-words assumption in large multi-topic texts may be somewhat mitigated by using the generative model to better capture the connections between terms during segmentation. Such models require that the words generated relate to both the segment and the subject. Therefore, the internal components of the document segments, not the full text, must be explicitly connected to the latent variables that model the topic. The segment-based generative model (SGM) that is suggested below explicitly takes into account the segments in each document by including segment model variables in the creation process. The SGM model is depicted graphically in figure 3.3.

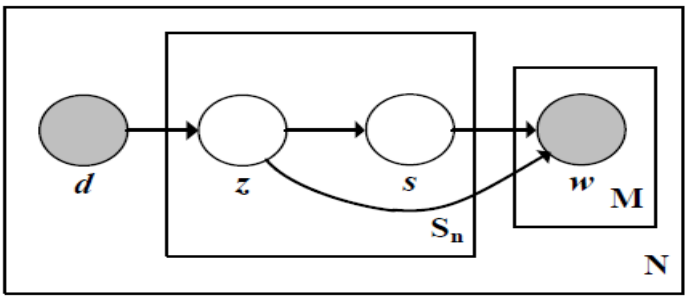


Figure 3.3. Plate notation for the SGM generative model.

Each document d ∈ D is seen by SGM as consisting of a set S_d of non-overlapping continuous text blocks or segments and a sequence of words n_d. The authors implemented their segmentation approach using text tiling, which is distinct from SGM. With SGM, model variables *s* and z are used to represent document segments and the distribution of model themes, respectively. The

following succinct statement summarises the SGM production procedure in corpus D of segmented documents:

1. Choose a document *d* from *D* through Probability (*d*)
2. In each segment *s* element of S_d
   a) Select a topic *z* for the document, where *d* ⇒ Probability (*z* | *d*).
   b) Is the corresponding segment the probability of segment *s, z* ⇒ Prob (*s* | *z*)?
   c) Loop through each word *w* in segment *s*.
      I. Select a word *w* from the inherent topic and segments, *w* ⇒ Pr (*w* | *z, s*).

SGM offers a more thorough document subject model that takes the text segment into account. The subject choice throughout the formation process (Pr (*z* | *d*)) is dependent on the topic's related probability (Pr (*s* | *z*)) in the segment, enabling a naturally occurring selection of topic affinity for each specific segment theme. Then, in addition to words being created by the subject, segments are also formed (Pr (*w* | *z, s*)). The aforementioned generating process may be converted into a triple made up of documents, segments, and words for each observation in a joint probability model for ternary data [246]:

$$\Pr(d, s, w) = \Pr(d) \sum_{z\, E\, z} \Pr(z|d) \Pr(s|z) \Pr(w|z, s). \quad (3.2)$$

3.3. Dimensionality Reduction

The input w (i, j) assigns the significance of the word "j" to the document "i" when representing a text collection as an array of document terms. Therefore, the amount of distinct items in a data collection determines its dimensionality. Even with medium-sized text data, it is typically extremely large—about 10,000. To differentiate between two provided learning vectors, the majority of learning algorithms employ some type of similarity metric. Due to the large dimensionality of the feature vector in the clustering of text documents, these similarity metrics lack discriminative ability. The feature vectors are nearly evenly distributed in a sparse high dimensional space, rendering conventional similarity assessments useless. The "curse of dimensionality" is used to describe this.

Many academics from many disciplines have developed various dimension reduction approaches in an effort to address the issue of high dimensionality. Two categories of dimension reduction techniques—feature transformation and feature selection—can be considered from a broad perspective on dimensionality reduction. By combining non-linear or linear combinations of the original dimensions, feature transformation algorithms aim to reduce the dimensionality of feature vectors to fewer new dimensions. In other words, the feature transformation approach projects the initial high-dimensional space to a lower-dimensional region. These techniques have

shown to be particularly effective in revealing latent structure in datasets via text clustering. Latent Semantic Analysis (LSA), Factor Analysis, Singular Value Decomposition (SVD), Independent Component Analysis (ICA), and Principal Component Analysis (PCA), are a few feature transformation methods that have been proposed in the literature. The goal of feature selection approaches is to exclude characteristics that don't seem to be useful for modeling, not to extract additional features. As a combinatorial optimization problem, this issue is represented.

The process of mapping data to a lower-dimensional space, which eliminates non-informative variations in the data or identifies the subspace in which the data is situated, is known as dimensionality reduction. Visualization, databases, statistics, text mining, data mining, machine learning, pattern recognition, optimization, and artificial intelligence are all areas of study in the multidisciplinary topic of dimensionality reduction. The following subsections address some of the primary methods for linear and nonlinear dimension reduction [244, 247].

3.3.1. Principal Component Analysis (PCA)

A type of linear dimensionality reduction technique is called principal component analysis (PCA). The aim of PCA is to create new variables that are an uncorrelated linear combination of the original variables. A portion of the underlying variables that characterize the data are discovered. PCA creates an uncorrelated projected distribution by minimizing the sum of squared errors or maximizing the variance and projecting n-dimensional data onto a smaller d-dimensional subspace. The majority of the time, data will have a sparse underlying structure. However, PCA frequently produces complex expressions that are challenging to decipher. It is calculated by applying the Eigen decomposition algorithm on the data covariance matrix, $\sum$. The most appropriate representations of the covariance matrix are the modularity matrix and the Laplacian matrix. The following elements make up the covariance matrix:

$$\sum = U \wedge U^T \tag{3.3}$$

Where U is the matrix having the relevant eigenvectors and $\wedge$ is the symmetric matrix carrying the eigenvalues along the diagonals. While the eigenvalues indicate the variance of projected inputs along primary axes, the resulting eigenvectors are comparable to the principal axes of the greatest variance subspace. The estimated dimensionality of the data is shown by the number of significant eigenvalues. Eigen decomposition is computationally costly for very high dimensional data since the volume of the covariance matrix is proportionate to the data dimensionality. Low-dimensional data are guaranteed to be accurately represented using PCA, which is simple to compute. On non-correlated data, it does not, however, offer a high level of accuracy.

3.3.2. Singular Value Decomposition

SVD is a dimensionality reduction method, which takes a set of highly variable and high-dimensional data points as input, reducing the dimensionality to a smaller dimensional space, thereby expressing the data substructure more clearly. SVD reduces a large matrix into a very small matrix based on the theorems of linear algebra. It yields a good estimate of the matrix's best rank k. Assume that X is a matrix r of rank $m \times n$. Let the eigen values of a matrix $\sqrt{XX^T}$ be $\sigma 1 \ldots \sigma r$. Then there is a diagonal matrix $S = diag\ (\sigma 1,\ \sigma r)$ and two orthogonal matrices, $U = (u1, \ldots\ldots, ur)$ and $V = (v1, \ldots\ vr)$, whose column vectors are symmetrical. The singular value decomposition of a matrix X is defined as $X = USV^T$ where the vectors $\sigma 1\ \ldots . \sigma r$ are its singular values. It is a scale factor that represents the relative significance of each new axis, and The column of V^T specifies the new axis, while the row of U denotes that the objects in the space are covered by the new axis. The rank r of the SVD of X, where r is the smaller of the two matrix dimensions, has at most r non-zero singular numbers. From here on, a k reduced singular decomposition of X is created using just the maximum k singular values. The least amount of reconstruction error and the best low-rank estimate are produced by SVD. The resulting matrix does not allow for an explanation of the different elements of the real data. The output of the SVD is always dense, regardless of the input's sparsity.

3.3.3. Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)

Similar to t-SNE, the dimension reduction method known as uniform manifold approximation and projection (UMAP) can be utilized for visualization as well as generic non-linear dimension reduction. Three presumptions regarding the data form the basis of the algorithm:

- On the Riemannian manifold, the data are evenly distributed.
- Locally, the Riemannian metric is consistent (or can be said estimated to be)
- There are connections within the manifold.

These presumptions allow one to construct a fuzzy topological model of the manifold. By looking for a low-dimensional projection of the data with the most comparable fuzzy topological structure, the embedding is discovered. UMAP generates initial "fuzzy simplicial complexes" to produce high-dimensional graphs. This is just a representation of a weighted graph, where the edge weights represent the probability that two points are connected, which is a means of constructing topological spaces from simple combinatorial components. This reduces the complexity of processing continuous geometry in topological space to relatively simple combinatorics and counting tasks. In order to ensure that all faces are present, the complex K is a collection of simplexes where each simplex's face is both in K and K + 1. Any two complexes in K that overlap each other form their respective faces. Large topological spaces can be constructed in this way.

UMAP stretches radii outward from each location and links the spots when those radii overlap to establish the connection. The selection of this radius is crucial. The cluster will be tiny and isolated if there aren't enough options, but everything will be connected if there are too many. This obstacle is solved by UMAP, which selects a radius locally depending on the distance to the nth closest neighbor of the point. UMAP then "Fuzzy" the graph by reducing the chances of a connection as the radius increases. Last but not least, UMAP stipulates that all points must be connected at least to their nearest neighbors, ensuring that the local structure is in harmony with the global structure [248–250].

3.4. Similarity Functions

When text documents are clustered, related documents will be grouped to form a unified group, while dissimilar documents will be separated into several groups. The definition of a pair of papers that are comparable or distinct, on the other hand, is not always apparent and typically varies depending on the precise nature of the issue. For instance, two papers with comparable subjects are grouped together when clustering research articles. When employing clustering on a website, we are frequently more interested in classifying the component pages according to the sort of data they present. For instance, when it comes to university websites, we could wish to differentiate between the homepages for teachers and students as well as between course pages and research project pages. The closeness between two items must be precisely defined, either by pairwise similarity or by distance, in order to do exact clustering.

The literature has a wide variety of clustering algorithms. The majority of them need knowledge of the similarities and differences between text items. Some of them need a definition of similarity and a representation of each item to be able to do computations when necessary. Different applications have different definitions of similarity and how objects are represented. Typically, building a representation and defining similarities based on it is convenient. This objective can be attained in several ways. For instance, the difference may be described as the separation between two objects if they can be expressed as coordinates in an n-dimensional vector space. The clustering method creates a partition that tries to match specific requirements when given a set of items and their similarities. It is likely best if the items in each category are as comparable as possible. However, in order for the findings to be applicable, there must be confidence that the concept of similarity accurately captures the similarities between actual items. Different distance or similarity measurements have been devised and are often utilized.

The basis of text cluster analysis techniques is determining how similar two things are. There are three primary processes to figuring out how similar two items are:

1. The selection of the variables to be utilized in object characterization.
2. Selecting a weighting system for these factors.

3. The selection of a similarity coefficient to assess how similar the two attribute vectors are to one another.

In order to do accurate clustering, the distance or pair-wise similarity between two items must be precisely defined. Numerous measurements of similarity or distance, including cosine similarity, Euclidean distance, the Jaccard correlation coefficient, Pearson correlation, and averaged Kullback-Leibler divergence, have been widely developed and used [248].

Several frequently used similarity measurements are:

- Euclidean Distance: Euclidean distance is a common unit of measurement for geometric issues. It is simple to measure with a ruler and represents the typical separation between two places. Additionally, the K-means algorithm uses it as the default distance measure.

$$dist(d_1, d_2) = \sqrt{\sum_{i=1}^{M}\left(d_1^i - d_2^i\right)^2} \tag{3.4}$$

where d_1^i corresponds to the ith element in the d_1 text document vector.

- Cosine Similarity: The correlation between the vectors indicates how similar two text documents are to one another. This is expressed mathematically as the cosine of the angle formed by the vectors, or cosine similarity.

$$cosine(U, V) = \frac{\sum_{i=1}^{k} f(u_i).f(v_i)}{\sqrt{\sum_{i=1}^{k} f(u_i)^2} \cdot \sqrt{\sum_{i=1}^{k} f(v_i)^2}} \tag{3.5}$$

- Jaccard Coefficient: The Jaccard coefficient contrasts the weights of terms that are included in both text documents but are not shared terms with those that do not. By considering its non-zero elements to be members of the set, it may be used to apply to feature vectors. According to this reasoning, the Jaccard coefficient calculates similarity by fusing the intersection of two texts and normalizing the result by fusing their union:

$$J(D_1, D_2) = \frac{|\, D_1 \cap D_2 \,|}{|\, D_1 \cup D_2 \,|} \tag{3.6}$$

Here, D_1 and D_2 are represented as sets of d_1 and d_2 documents together.

- Pearson Correlation Coefficient: It provides a further indicator of how closely two vectors are connected.

$$P(d_1, d_2) = \frac{\sum_{i=1}^{M}(d_i^1 - \bar{d})\,(d_i^2 - \bar{d})}{\sqrt{\sum_{i=1}^{M}(d_i^1 - \bar{d})^2 \;\sum_{i=1}^{M}(d_i^2 - \bar{d})^2}} \tag{3.7}$$

Where d_1^i is the ith element in the d_1 document vector and $\bar{d}$ is the mean of the values of the d_1 document vector.

- Averaged Kullback-Leibler Divergence: Relative entropy, often known as the Kullback-Leibler divergence (KL divergence), is a commonly used metric for assessing the distinctions between two probability distributions.

$$KL(d_1, d_2) = \sum_{i=1}^{M} d_i^1 \log\left(d_i^1 \Big/ d_i^2\right) \tag{3.8}$$

3.5. Machine Learning Topic Models and Text Clustering Algorithms

There are many different categories into which text clustering algorithms can be categorized, including partitioning algorithms, agglomerative clustering algorithms, grid-based algorithms, model-based algorithms, density-based algorithms, frequent pattern-based algorithms, common parametric modeling-based techniques like the Expectation-Maximization algorithm (EM-algorithm), and constraint-based algorithms.

Partition clustering creates a set of data subsets or clusters with no overlap so that each data object is in a unique subset. In order to make these breakthroughs, a value for the desired number of preferred clusters must be chosen. K-means and its version bisecting K-means, PAM, K-medoids, CLARA, CLARANS, and other public heuristic clustering approaches are a few examples.

A nested collection of clusters arranged as a tree is produced using the hierarchical clustering algorithm. Agglomerative hierarchical clustering and divisive hierarchical clustering algorithms are possible iterations of the hierarchical clustering algorithm. The bottom-up algorithms, which initially regard each item as a split cluster and progressively merge a pair of neighboring clusters to form new clusters until all of the clusters are combined into one, are described as agglomerative algorithms. Chameleon, UPGMA, BIRCH, and ROCK are examples of public hierarchical approaches.

The data items are grouped in arbitrary forms using density-based clustering techniques. Density determines clustering (number of objects). DENCLUE, DBSCAN and its variations, and OPTICS are the density-based open - sourced approaches.

The data items are gathered using a multi-resolution grid structure in the grid-based clustering approach. The quick processing time of this approach is a benefit.

Model-based approaches fit the data to a specific model after using a model for each group. Additionally, it is used to automatically count the number of clusters.

Models selected from a portion of dimensions are used in pattern-based frequent clustering to gather the data items. Techniques for clustering under constraints are those that depend on user- or application-specific limitations. Users must provide limits on clusterings, such as user needs or an explanation of the characteristics of the clustering findings, in order to do this [26].

In this Thesis, several commonly used text clustering algorithms used for text clustering will be discussed. Special focus will also be paid to text clustering algorithms for social media text data, web-based text data, and dynamic data scenarios.

3.5.1. Distance-based Clustering Algorithms

Algorithms for distance-based clustering are created by measuring the closeness of text items using similarity functions. The selection of the metric is crucial for text clustering because there are many different distance metrics available. The cosine similarity function is the most well-known distance-based similarity metric frequently applied in the text domain. Assume that the vectors of normalized and damped frequency terms in the two distinct documents U and V are $U = (f(u_1) \ldots f(u_k))$ and $V = (f(v_1) \ldots f(v_k))$. The (normalized) frequency term is represented by the values $u_1 .. u_k$ and $v_1 .. v_k$ while the damping function is represented by the function $f(\cdot)$. Either the square root or the logarithm can be used to define the usual damping function of $f(\cdot)$. A fundamental issue in information retrieval is determining text similarity. Many weighted heuristics and similarity functions may also be used for text clustering, despite the fact that the majority of information retrieval research focuses on how to assess the similarity between keyword searches and text documents rather than the similarity between two articles.

3.5.1.1 Distance-based Agglomerative and Hierarchical Clustering Algorithms

In the clustering literature, hierarchical clustering methods have been thoroughly examined for a variety of records, including categorical data, text data, and multivariate numerical data. Because it naturally generates a hierarchy in the form of a tree that can be employed in the search process, the agglomerative hierarchical grouping approach is particularly helpful for supporting numerous search methods. Agglomerative clustering's fundamental idea is to combine documents into groups based on how similar they are to one another. On the basis of the greatest pairwise similarity between these groups of documents, almost all hierarchical clustering algorithms constantly combine groups.

How these algorithms determine this pairwise similarity between various groups of texts is the primary distinction between them. The best-case, average-case, or worst-case similarity between the documents collected from these group pairings, for instance, may be used to determine how similar two groups are. Conceptually, the process of grouping documents into consecutive top-level clusters produces a dendrogram or cluster hierarchy, where leaf nodes represent individual documents and interior nodes represent collections of merged clusters. When two clusters are combined, a new node in this tree is made to represent the bigger combined group. The two document sets that have been combined into it are represented by the two child nodes of this node. The different methods of merging document groups for different agglomerative methods are:

- Single-linkage clustering
- Group-average linkage clustering
- Complete-linkage clustering

3.5.1.2 Distance-based Partitioning Algorithms

In the literature, clusters of items are effectively created using partitioning methods. Below is a discussion of the two most popular distance-based partitioning techniques.

Algorithm for clustering K-medoid: With the k-medoid clustering approach, we create clusters around a collection of anchor points (or medoids) from the original data. The cluster is constructed around the corpus, and the fundamental objective of the method is to identify the group of texts that most accurately replicates the original corpus. Each document is assigned to the collection's closest match. As a result, a collection of ongoing clusters are produced from the corpus and are continually improved by random processes. The technique employs an iterative process, where the list of k representatives is continually enhanced by random exchanges. We specifically identify the goal function that requires improvement in this exchange process as the mean similarity among each document in the corpus and its nearest representation. If the clustering objective function improves, we replace the randomly chosen sample in the current medoids point set at each iteration with a randomly chosen sample from the set; this is done until convergence is reached.

A collection of k representatives are also used by the k-means clustering method to create clusters based on these representatives. These illustrations, however, deviate somewhat from the k-medoids technique and are not always based on the actual data. The simplest version of the k-means approach distributes documents to a set of k seeds in the initial corpus depending on their degree of similarity. The seed from the previous iteration is replaced in the following iteration by the centroid of the point allotted to each seed. In other terms, the new seed is described as this cluster's superior center. This process is repeated until convergence. The fact that the k-means approach takes less iterations to reach convergence than the k-medoids method is one of its benefits. Five iterations or less seem to be adequate for effective grouping for many big data sets, according to observations from research applications. The k-means method's key drawback is that it is still extremely dependent on the initial seed set gathered during clustering. Second, a particular document cluster's centroid may include a sizable number of words. The similarity computation in the following iteration will take longer as a result. One method for enhancing the quality of the initial seeds gathered for the clustering process is to determine the initial seed set using another light clustering approach, such as the agglomerative clustering technique, and to monitor the initial seed creation process using some kind of partial supervision.

3.5.2. Probabilistic Document Clustering and Topic Models

Topic modeling is a prominent approach to probabilistic document clustering. A probability generative model for corpora text documents is what topic modeling is intended to achieve. The fundamental approach is to model a corpus using hidden random variables, whose

properties are determined using a particular document collection. Any topic modeling method's fundamental random variables (and associated tenets) are as follows:

- Supposing that there is a chance that each of the k topics will include the probability of at least *n* of the corpus's documents. As a result, a given document may be more likely to belong to more than one subject, reflecting the fact that a single document may cover more than one topic. the probability that the document *Di* belongs to a certain topic *Tj* is given by *P (Tj | Di),* for a given document *Di* and a set of topics *T1. .. Tk.* We see that the themes resemble clusters in nature, and the value of *P (Tj | Di)* indicates the likelihood that a given text will belong to a certain cluster, from the ith cluster to the jth cluster. Since the membership of the documents in a non-probabilistic cluster is intrinsically predictable, the cluster is typically a tidy division of the document collection. However, when document themes transcend different clusters, this frequently presents problems. An excellent approach to solving this problem is to use probabilistic soft cluster members. Finding probable subject clusters in the underlying text collection is the primary objective in this example, with establishing the document's participation in the group serving as a secondary objective. Consequently, topic modeling is the name of this academic discipline. Although it is connected to clustering issues, it is often investigated as a separate study area. One of the key outcomes of the process is an estimation of *P (Tj | Di)* utilizing the topic modeling approach. Similar to the number of clusters, the value of k is one of the inputs to the algorithm.
- A probability vector that measures the likelihood that certain phrases in the subject dictionary will appear is linked to each topic. Let the *d* words in the lexicon be t_1.....t_d. The likelihood that the phrase t_l will appear in a text that is exclusively about subject T_j, is therefore given by *P* (t_l | T_j). Another crucial parameter that has to be estimated utilizing topic modeling techniques is the value of *P* (t_l | T_j). Another crucial parameter that has to be estimated utilizing topic modeling techniques is the value of *P* (t_l | T_j).

Keep in mind that *n* stands for the quantity of documents, *k* for the topic, and *d* for the size of the dictionary (word). To modify the probability of a given corpus of documents as big as feasible, most topic modeling techniques try to employ the maximum likelihood method to learn the aforementioned parameters. Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Index (PLSI) are the two fundamental techniques for modeling topics [184].

3.5.2.1 Latent Dirichlet Allocation (LDA)

LDA is the most popular topic model for assigning topics to document sets. David Blei, Andrew Ng, and Michael Jordan (Blei et al. (2003) developed the LDA technique. Although LDA is a generative model, text mining, introduces a way to attach topical content to text documents.

Each document is viewed as a combination of several distinct topics. An advantage of the LDA technique is that neither the number of topics nor what the topics will look like is known in advance. LDA can be used to determine trends over time by identifying “hot” and “cold” topics. By tuning the LDA parameters to fit different dataset shapes, topic formation and resulting document clusters can be explored.

LDA treats a document as a set of probability distributions over words or topics. These themes are not strictly defined as they are identified based on the probability of co-occurrence of the words they contain. To get ranked lists of words associated with a given word w_n, the set of topics generated by LDA are taken, and then for each word contained, the sum of the weight of each topic is multiplied by the weight of the given word w_n in this topic.

Formally, for N topics and w_{ji} that indicate the weight of word i in topic j, the rank weight of word i is calculated as follows:

$$w_i = \sum_{j=1\ldots N} w_{ij} * w_{nj} \tag{3.9}$$

This representation allows for the ranking of words associated with a particular word w_n, based on the probabilities of co-occurrence in the document [252]. However, LDA has two major limitations: 1) it assumes the text is a mixture of topics, and 2) it performs poorly on short text (<50 words).

3.5.2.2 Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM)

GSDMM, which is a short form for Gibbs Sampling Dirichlet Multinomial Mixture, is a model proposed by (Jianhua Yin and Jianyong Wang, (2014)) that works better than vanilla LDA. It assumes that one document is only about one topic, and it works great on short texts like tweets and movie reviews. In contrast to LDA, GSDMM just requires a threshold (maximum number of topics), and the actual number of topics is determined by the data. This indicates that the GSDMM Algorithm is capable of automatically determining the number of clusters. The depth and uniformity of the clustering findings may be balanced clearly using GSDMM. In contrast to techniques based on the Vector Space Model (VSM), GSDMM converges quickly with careful parameter selection. The sparse and high-dimensional issue of brief texts can be handled with GSDMM. GSDMM may also retrieve the relevant words of each cluster, just as other topic models (such as PLSA and LDA) [207].

3.5.2.3 TopicBERT

A topic modeling approach called TopicBERT or BERTopic uses BERT embeddings (Bidirectional Encoder Representations from Transformers) and a class-based TF-IDF to produce dense clusters that make subjects simple to understand while preserving key terms from the topic descriptions. It belongs to the Contextualized Topic Models (CTM) family of topic models, which

employ pre-trained linguistic representations (like BERT) to facilitate topic modeling as shown in figure 3.4 below.

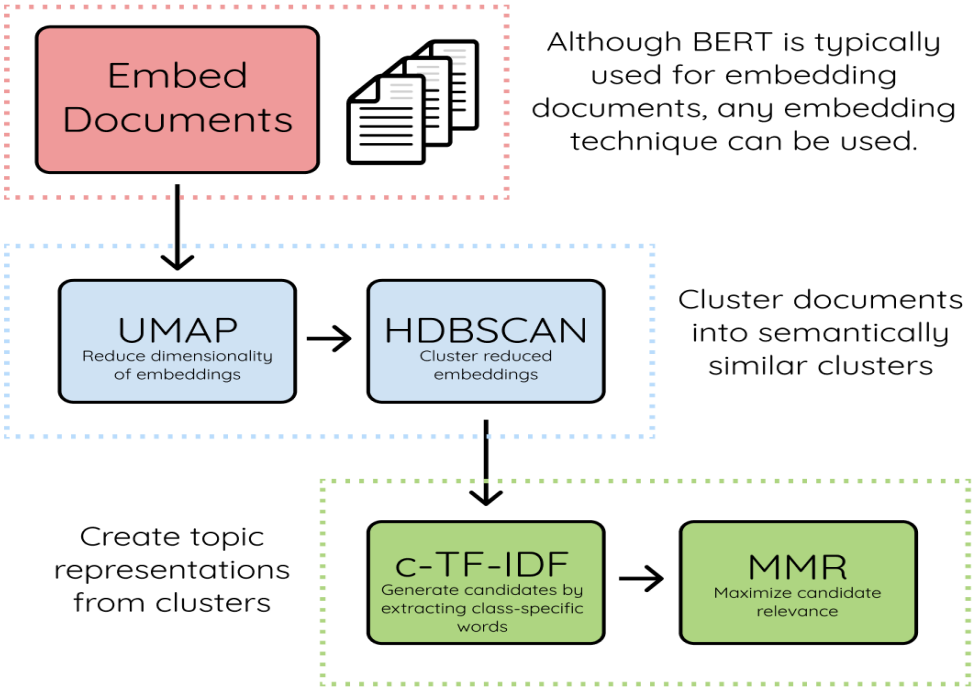


Figure 3.4. BERTOPIC / TOPICBERT representation.

The algorithm consists of three stages:

- Embed text data (document).

  In this step, the algorithm uses BERT to extract the document embedding. Alternatively, other embeddings and embedding techniques can be used. By default, the following sentence transformers are used.

  1. "paraphraseMiniLML6v2" This is an English BERT-based model specially trained for semantic similarity tasks.
  2. "paraphrasemultilingualMiniLML12v2" This is similar to the first one, with one major difference the xlm model works in over 50 languages.

- Cluster document

  Use UMAP to reduce the dimensions of the embedding and use the HDBSCAN technique to group the reduced embeddings to create a cluster of semantically similar documents.

- Create a topic presentation

  The final step is to use class-based TFIDF to extract and reduce topics and improve word consistency with maximum marginal relevance.

It can be deduced that the algorithm combines text-clustering techniques to create topic models that are representatives of the clustered text documents. Topic creation is done using a class-

based variant of TF-IDF (c-TF-IDF) to compare the importance of words between documents. The intuition is shown below:

$$c - TF - IDF_i = \frac{t_i}{w_i} * \log \frac{m}{\sum_j^n t_j} \tag{3.10}$$

Where each word's frequency t is determined for each class i and divided by the total number of words (w). This is a method of regularizing terms that are used often in the class. The entire number of documents, m, is divided by the overall occurrence of words, t, across all classes, n. The top 20 words for each subject are selected based on their c-TF-IDF scores to produce a topic representation. Since the score is a measurement of information density, the higher the score, the more representative it should be of its topic. There is a chance that, depending on the size of the dataset, hundreds of topics were created. The topic reduction can be performed by tweaking the parameters of HDBSCAN such that fewer topics can be gotten through its min_cluster_size parameter. It is worthy of note that adjusting the HDBSCAN parameters does not allow for the specification of the exact number of clusters [253].

3.6. Evaluation of Topic Models and Text Clustering Algorithms

The assessment of clustering findings is one of the most crucial components of cluster analysis. Analyzing the output to see how accurately it replicates the data's original structure is called evaluation. Three sections make up the assessment form:

- Internal quality measurement: Since no outside information is used, the generic similarity measurement is applied in this case based on how similar the matched documents are. One may gauge how similar two clusters are by looking at their cohesiveness. Utilizing the weighted intra-cluster similarity as a measure of cluster cohesiveness is one technique.
- External quality measurement: It entails contrasting the cluster analysis findings with outcomes that are externally known, such as class designations that are given by a third party. It gauges how well the cluster label complies with a class label supplied externally. This method is mostly used to choose the best clustering technique for a certain dataset since the number of "true" clusters is known in advance.
- Relative cluster measurement: Different hyper-parameters for the same method are changed to assess the clustering results (e.g. changing the number of clusters in a k-means algorithm).

3.6.1. Silhouette Coefficient

The evaluation of a clustering result must be done using the model itself when the ground truth tags for a dataset are unknown. One such evaluation method is the Silhouette Coefficient or Silhouette score, where a high Silhouette Coefficient score indicates a model with clearly

identifiable clusters. To determine how effective a clustering method is, one might utilize a measure called the silhouette Coefficient. Its value is between -1 and 1. For every sample, the Silhouette Coefficient is specified, and it consists of two scores:

- The typical distance separation between a sample and every other point in a class.
- the average separation separating a sample from every other point in the adjacent cluster.

Then, a single sample's Silhouette Coefficient is provided as:

$$S = \frac{b-a}{\max(a,b)} \tag{3.11}$$

The mean of the silhouette coefficient for each sample makes up the silhouette coefficient for a group of samples.

Where a = average intra-cluster distance (the average distance between each point within a cluster).

b = average inter-cluster distance (the average distance between all clusters).

3.6.2. Calinski-Harabasz Index

The Calinski-Harabasz index, commonly known as the variance ratio test, can be used to assess the model in cases when the ground truth label is unknown. A model with a more favorable Calinski-Harabasz score in this case has a more clearly defined cluster. The ratio of all clusters' within-cluster variance to the total of variances across clusters, where variance is calculated on the basis of squared distances, is the index. The Calinski-Harabasz score *S* is the ratio of the between-cluster dispersion mean to the within-cluster dispersion for a collection of data *E* of size n_E that has been clustered into k clusters:

$$S = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k-1} \tag{3.12}$$

Where $tr(W_k)$ denotes the trace of the within-cluster dispersion matrix and $tr(B_k)$ denotes the trace of the between-group dispersion matrix defined by:

$$W_k = \sum_{q=1}^{k} \sum_{x \in c_q} (x - c_q)(x - c_q)^T \tag{3.13}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{3.14}$$

With c_q the collection of points in cluster q, c_q the center of cluster q, c_E the center of E, and n_q the number of points in cluster q.

3.6.3. Davies-Bouldin Index

A lower Davies-Bouldin index refers to a model with higher cluster separation, which may be used to evaluate the model if the ground truth labels are unknown. This index represents the average degree of "similarity" between clusters, where "similarity" is a metric that contrasts the

separation between clusters with the size of the clusters. The lowest score that can be given is zero. Values nearer to 0 suggest a more effective division [254].

The index is the mean similarity between the most comparable cluster $C_{j.}$ and each cluster C_i for i= 1, …….., k. The similarity is described in the context of this index as a trade-off metric R_{ij}:

- s_i, commonly known as cluster diameter, is the arithmetic mean between each point in cluster i and its centroid.
- d_{ij}, which is the separation distance between cluster centroids j and i.

A straightforward option to create R_{ij} that is symmetric and nonnegative is:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{3.15}$$

The Davies-Bouldin index is therefore described as:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max R_{ij} \tag{3.16}$$

3.6.4. Coherence Scores

Topic modeling coherence scores are used to measure how humanly interpretable a topic is. In this case, the topic appears as the top N words most likely to belong to that particular topic. In other words, the coherence score measures how similar these words are to each other. The four-stages pipeline is:

- Segmentation
- Probability Estimate
- Confirmation Measurement
- Aggregation

One of the most popular coherence metrics is called CV. It builds a word content vector using the general appearance of the word and calculates the score using the normalized cosine similarity and point-by-point mutual information (NPMI). This metric is popular because it is the default metric for the Gensim topic coherence pipeline module, but it has some problems. Other coherence metrics in literature are UMass coherence Score, UCI coherence Score, word2vec coherence score, etc [255].

3.7. Deep Learning Text Clustering Algorithms

Text clustering confronts a number of difficulties, including complicated semantics, a wide feature space, and high dimensionality, all of which may reduce the accuracy of the grouping since certain aspects may be unimportant. The way clustering is described is another issue. One of the biggest issues with conventional algorithms is this. Clustering results cannot be conceptually described. Deep neural networks' innate capacity to do non-linear conversions allows them to

transform data into cluster-friendly representations, in contrast to typical linear feature conversion techniques. One method for resolving this issue and obtaining the most precise document clustering outcomes is deep learning. Deep learning clustering is the main topic of recent writing (also called deep ensemble clustering or deep clustering).

Without overly depending on human-engineered characteristics, deep learning has also been frequently utilized to build better and richer data representations from massive data. Many of these deep representation networks are built on an initial stage of unsupervised learning known as unsupervised pre-training (e.g., restricted Boltzmann machines, matrix factorization, AutoEncoders, etc.), which trains the networks to learn more accurate and detailed representations of the data. This has been demonstrated to significantly enhance the outcomes of supervised learning activities in real-world settings. The success of deep learning has lately inspired various deep learning-based advancements of clustering algorithms that are immediately incorporated into unsupervised learning, despite the fact that it primarily originated in the field of supervised learning. In order to enhance the quality of the clustering results and make clusters simpler to extract, the majority of deep learning based clustering algorithms have taken advantage of the representational power of deep learning networks for preprocessing clustering inputs.

3.7.1. Working Principles of Deep Learning-based Clustering methods

Let's have a look at the issue of grouping n samples, X = {x(1), x(2),...,x(n)} into K-categories, each denoted by a centroid μj, j=1,…, K where X ∈ RD. The pipelining technique uses a deep learning-based clustering algorithm that is typically trained in two steps and operates on basically identical operational principles:

- Phase 1: Use the DNN architecture for parameter initialization and representation learning, and use non-clustering loss(for example, standard representation learning) for training. Extraction of latent characteristics (LFs), a kind of cluster-aware data representation, from one or more layers follows (based on the network architecture type).
- Phase 2: The cluster assignment is formed, and then the parameters are optimized by iteratively computing the secondary target distribution and reducing the cluster loss, such as the Kullback-Leibler divergence (KLD) and cluster assignment hardening loss (CAHL). When a machine learning-based clustering technique is used to iteratively improve the clustering target, backpropagation updates the centroid. K-means and agglomerative clustering algorithms in particular are often employed in the literature.

As a result, samples are changed by nonlinear mapping f_θ: $X \rightarrow Z$, where θ is a learnable parameter and $Z \in R^K$ is the learnt or feature embedded space, where K≪ D. This technique prevents samples from being clustered in the original input space X directly. Due to their function approximation qualities and feature learning capabilities, Deep Neural Network (DNN)

architectures like AutoEncoders (AEs) are used to parametrize f_θ. However, in phase 2, the network is often trained and updated to minimize both non-clustering and clustering losses simultaneously for a superior clustering outcome [8, 166, 233].

3.7.2. Deep Clustering Framework

The deep neural network, network loss, and clustering loss are the three fundamental parts of the deep clustering method.

3.7.2.1 Deep Neural Network Architecture

The representation learning element of the deep clustering method is a deep neural network. From datasets, they are used to create low-dimensional non-linear data representations. Although generative models like generative adversarial networks and variational autoencoders have also been employed in other methods, autoencoders are still the most commonly used architecture. Convolutional neural networks, which alter the network architecture, are also often utilized. Figure 3.5 below illustrates a typical deep clustering architecture.

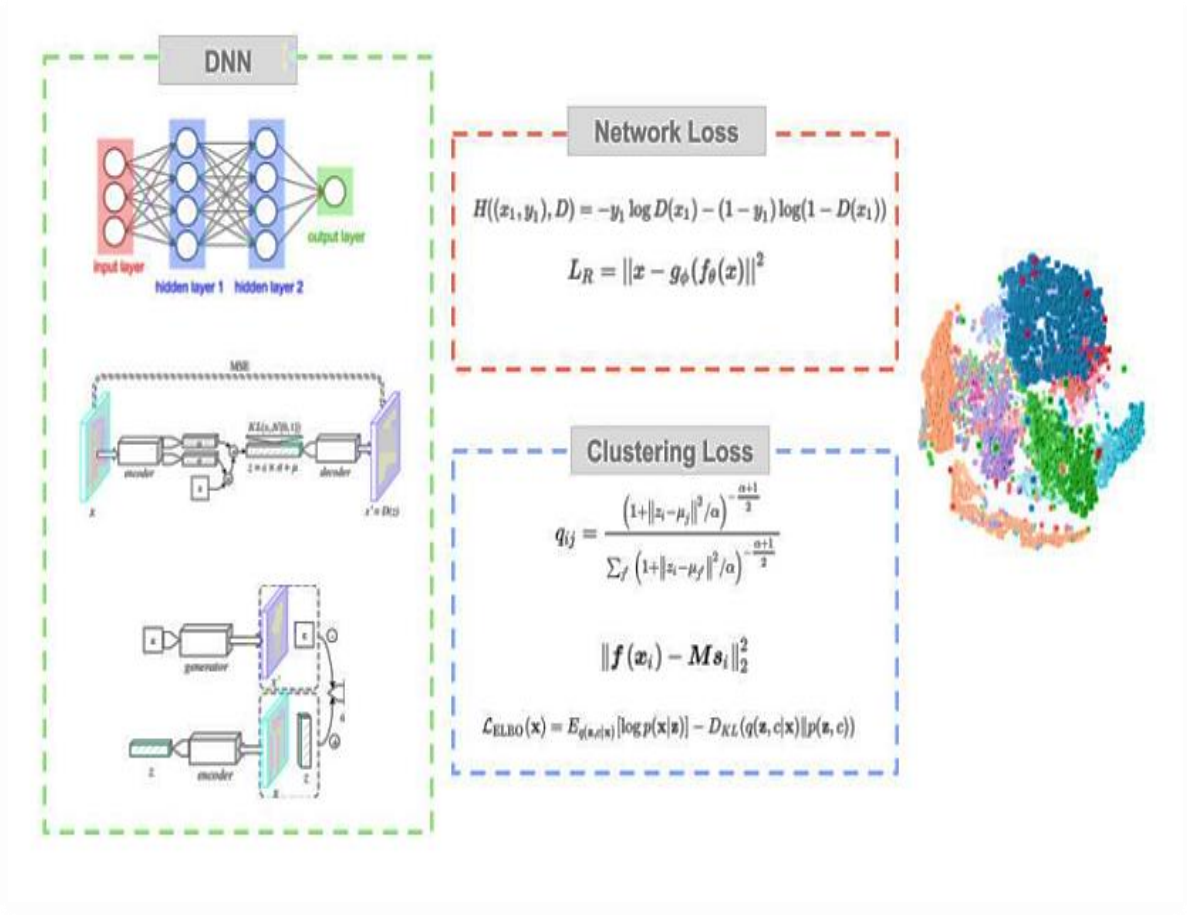


Figure 3.5. Deep Clustering Framework.

3.7.2.2 Loss function

The cluster-oriented loss L_R and network loss L_C are two unsupervised representation learning losses that are often combined linearly to form the deep clustering algorithm's objective function. They have the following form:

$$L = \lambda L_R + (1-\lambda)\, L_C \tag{3.17}$$

Where λ is a hyper parameter between zero and one, used to balance the effects of the two loss functions.

- **Network loss**

Loss of variational loss of VAE, AutoEncoder reconstruction, or adversary loss of GAN is all examples of neural network losses. For the initialization of deep neural networks, network loss is crucial. Usually, the clustering loss is generated by altering the hyper parameter λ after a few epochs. Some models, such IMSAT, DAC, and JULE, fully forgo network loss in favor of solely using pooling loss to direct representation learning and pooling.

- **Clustering Loss**

In different techniques, different clustering losses have been proposed in the literature. The losses may be broadly divided into two categories:

I. Cluster Assignment: There is no need to perform another clustering algorithm on the representation of learnt data because cluster assignment losses immediately offer cluster assignments for data points. Agglomerative clustering loss, cluster assignment hardening loss, and K-means loss are a few examples.

II. Cluster Regularization: On the other hand, cluster regularization loss merely compels the network to keep the proper discriminatory information derived from the data in the representation. To obtain the clustering result, more clustering on the representation space is needed. Examples include locality preserving loss and group sparsity loss.

3.7.3. Deep Clustering Architecture

Deep clustering techniques are based on various network structures and the type of loss function being used. The following three categories might essentially classify these architectures:

1. AutoEncoders models.
2. Generative models.
3. Direct cluster optimization models.

The AutoEncoder-based models are discussed below. The representation of an AutoEncoder is illustrated in figure 3.6 below.

3.7.3.1 Based on AutoEncoders

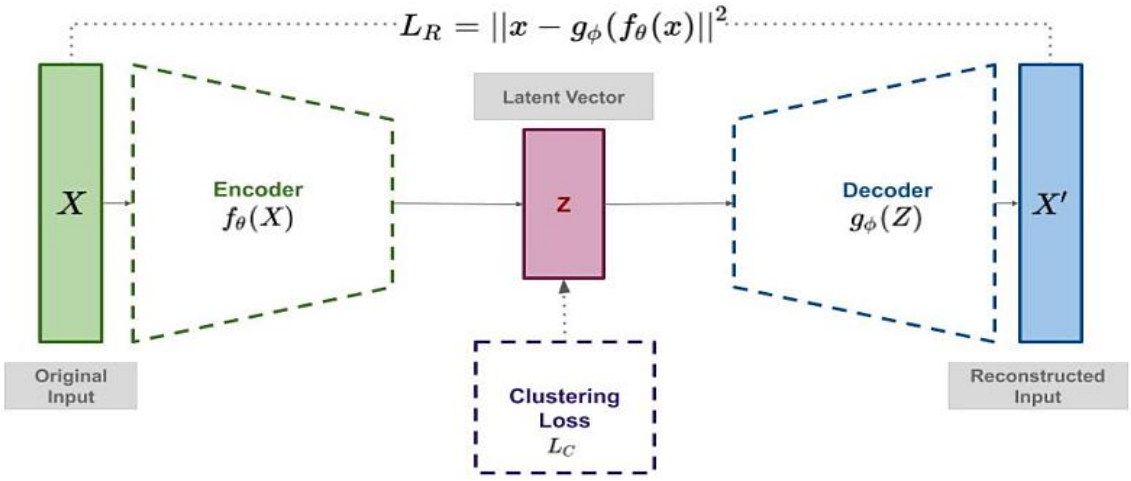


Figure 3.6. AutoEncoder-based Deep Clustering

From denoising to neural machine translation, autoencoders have been widely employed in unsupervised representation learning problems. Deep clustering algorithms frequently employ this straightforward yet incredibly effective foundation. The reconstruction loss is used to set the encoder and decoder metrics before the clustering loss is added in the majority of AE-based deep clustering algorithms. Different low-dimensional data generating distributions are learned by the AutoEncoder. Although there is a lot of group overlap, the representation space is condensed. To appropriately distinguish the class's submanifolds from the clusters, the representation space must be regularized. The feature space is destroyed as a result, which reduces the decoder's capacity to reconstruct. The embedding space *E*, which the representational network learns from the latent feaature space *Z* of the encoder, is a distinct representation space that may be used to avoid this trade-off. The parameterization of t-SNE served as the source of inspiration for this network.

First, the paired probability *P* denotes that the Student's t-distribution kernel is used to determine the likelihood of two points being combined in the *Z* encoded space:

$$P_{ij} = \frac{\left(1+\frac{\left\|f(x_i)-f(x_j)\right\|^2}{a}\right)^{-a+1/2}}{\sum_{k\neq l}\left(1+\frac{\left\|f(x_k)-f(x_l)\right\|^2}{a}\right)^{-a+1/2}} \tag{3.18}$$

The Student t-distribution is employed since there is no kernel width parameter and it approximates the Gaussian distribution to obtain greater degrees of freedom. Additionally, tighter probabilities are assigned. There is more room to simulate the spatial patterns of the representation space since the degree of freedom is thought of as being twice as large as the *Z* dimension. In the same way, the paired probability *q* denotes the likelihood that two points in the insertion space *E* are adjacent.

$$q_{ij} = \frac{\left(1+\left||h(z_i)-h(z_j)\right|\right|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\left||h(z_k)-h(z_l)\right|\right|^2\right)^{-1}} \tag{3.19}$$

This distribution is near to *p* by providing a strong repulsive force between the clusters to reduce the KL divergence. The degree of freedom is set to one, which restricts the degree of freedom to simulate the local structure. The cross-entropy of *p* and *q* is used to train the representation network, and it also has the effect of reducing the entropy of the *p* distribution.

- **Deep Embedded Clustering (DEC)**

One of the first deep clustering works, Deep Embedded Clustering is frequently used as a benchmark to assess how well other models perform. Cluster allocation hardening loss and AE reconstruction loss are used by DEC. The definition is based on the tSNE distribution and the degree of the free α cluster as one, to further refine the distribution of the soft cluster distribution q, and also define the distribution p_{ij}, which is the auxiliary target distribution derived from the update after each iteration of T.

$$q_{ij} = \frac{\left(1+\|z_i-\mu_j\|^2/a\right)^{-a+1/2}}{\sum_{j'}(1+\|z_i-\mu_j\|^2/a)^{-a+1/2}} \tag{3.20}$$

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}} \tag{3.21}$$

The initialization of the encoder and decoder settings during certain times of reconstruction loss occurs during the pre-training phase before the training begins. The encoder network is fine-tuned by maximizing the KL divergence between the soft clustering distribution q_{ij} and the auxiliary distribution p_{ij} after the decoder network has undergone pre-training. While iteratively assigning groups, this training may be seen as a self training process to improve the representation.

$$\boldsymbol{min} \sum_i \sum_j p_{i_j} \log \frac{p_{i_j}}{q_{i_j}} \tag{3.22}$$

- **Deep Clustering Network (DCN)**

An AutoEncoder is used by Deep Clustering Network to train a representation that is compliant with the K-means method. It optimizes k-means and joint reconstruction losses via alternate cluster allocation after pre-training the AutoEncoder. When compared to other techniques, the clustering loss of k-means is fairly understandable and straightforward. DCN identifies these as its objectives:

$$\boldsymbol{min} \sum_{i=1}^{N} = \left(\ell\left(g\left(f(x_i)\right), x_i\right) + \frac{\lambda}{2} \parallel f(x_i) - Ms_i \parallel_2^2\right) \tag{3.23}$$

Where *g* and *f* are decoder and encoder functions respectively, s_i is the assigned vector for data point *i*, which has only a single non-zero element, and M_k represents the centroid of the kth cluster [256].

4. RESEARCH DESIGN

This chapter presents the specific materials and methods used in this research to perform topic modeling and text document clustering of the COVID-19 vaccine dataset.

4.1. Data Collection

A recently identified strain of the coronavirus, a kind of virus known to cause respiratory infections in people, is what causes COVID-19, an infectious illness. Prior to December 2019, when a pneumonia outbreak with an unknown origin appeared in Wuhan, China, this new strain was unknown.

Since the Covid-19 pandemic, there has been a lot of discussion on the necessity for a COVID-19 vaccine on social media platforms and news websites due to the sharp rise in the number of Covid-19 cases. To provide light on COVID-19 vaccine administration statistics from August 2020 to February 2022, Data was obtained through Kaggle, a platform for competitions where businesses may publish actual problems that they have been having for a long time and make their data available for data scientists to work on. Data scientists and other developers may store datasets, organize machine-learning competitions, and build and share code using Kaggle. The types of data science problems posted on Kaggle can be anything from attempting to predict cancer occurrence by examining patient records to analyzing sentiment evoked by movie reviews and how this affects audience reaction. Different sources post projects on this trailblazing platform. While some are just for educational purposes and fun brain exercises, others are genuine issues that companies are trying to solve. Kaggle makes the environment competitive by awarding prizes and rankings for winners and participants. The prizes are not only monetary but can also include attractive rewards such as jobs or free products from the company hosting the competition. The data set obtained from Kaggle for this Thesis research contains Twitter tweets made with the hashtag #CovidVaccine.

4.1.1. Data Content

Using the hashtag #CovidVaccine, the dataset includes tweets from all over the world. The collection started on 1/8/2020 and ended on 10/2/2022. Some of the vaccines contained in the tweets include:

- BioNTech/ Pfizer
- Moderna
- Oxford/AstraZeneca
- Sinovac
- Sinopharm

- Sputnik V
- Covaxin

4.1.2. Data collection technique

Twitter's application programming interface - (API) was used to gather the information. The tweepy package was used in conjunction with a python script to extract the tweets. The daily updates to the data collection are in the public domain. For information on the data extraction process, please visit the following links for the data extraction script and pocketbook: https://www.kaggle.com/kaushiksuresh147/twitter-facts-extraction.

Information regarding the data set

There are 13 columns and 368,212 distinct records in the dataset. Table 4.1 below shows the description of the attributes:

Table 4.1. Data set features and their description.

No	Columns	Descriptions
1	User-name	The user's name as it appears on Twitter.
2	User-location	The location that the user specified for an account profile.
3	User-description	The account's UTF-8 string description, as defined by the user.
4	User-created	Date and time of the account creation.
5	User-followers	the current amount of followers a certain account had.
6	User-friends	The amount of friends currently linked to an account.
7	User-favourites	The current count of favorites for a certain account's posts.
8	User-verified	shows whether or not the user account has been validated.
9	Date	Date and time the COVID-19 tweet was published in UTC.
10	Text	The tweet's real UTF-8 string content about the coronavirus vaccine
11	Hashtags	Along with #CovidVaccine, all the other hashtags used in tweets about vaccines
12	Source	The utility or device used in posting the tweet.
13	Is-retweet	Indicates whether another authenticating user retweeted the tweet.

Only the text component of this dataset is the focus of this research project. The goal is to apply unsupervised learning algorithms and approaches to comprehend what Twitter users are talking about the ongoing COVID-19 vaccination distribution.

4.2. Data Analysis

The coronavirus vaccination tweets were subjected to text grouping and topic modeling on Google Colaboratory. Colab, often known as "Colaboratory," is an interactive platform that enables simple notebook sharing, free access to GPUs, and the authoring and execution of Python scripts using a browser. AI researchers, data analysts, data scientists, data engineers, and students can all

collaborate more effectively with Colab. Colab notebooks, like Jupyter notebooks, let you combine rich text and executable code in one document together with graphics, HTML, LaTeX, and other elements. Colab notebooks are first stored in the Google Drive account. The system configurations of the Colab pro virtual environment used in this research are shown below:

- vCPU : 2-core Intel(R) Xeon(R) CPU @ 2.30GHz with Level 3 cache
- RAM : 25GB RAM
- HDD : 225GB HDD with 75GB used and 150GB available
- GPUs: Tesla K80, P100, T4

4.2.1. Experimental Procedure for Text Document Clustering

The key processes for carrying out any text document clustering task are depicted in the figure 4.1 below. Preprocessing is done on the text documents before they are transformed into a document-term matrix. To end the dimensionality curse and lessen the sparsity of the document-term matrix, dimensionality reduction is used. In order to properly capture the semantics and relationships between words, document embeddings are then presented. Through document encoding and transformation, the text document and its word embedding are encoded to create a single document matrix. The new document embedding is then subjected to a variety of deep learning and machine learning techniques to create clusters with a high intra-cluster similarity and a low inter-cluster similarity [257].

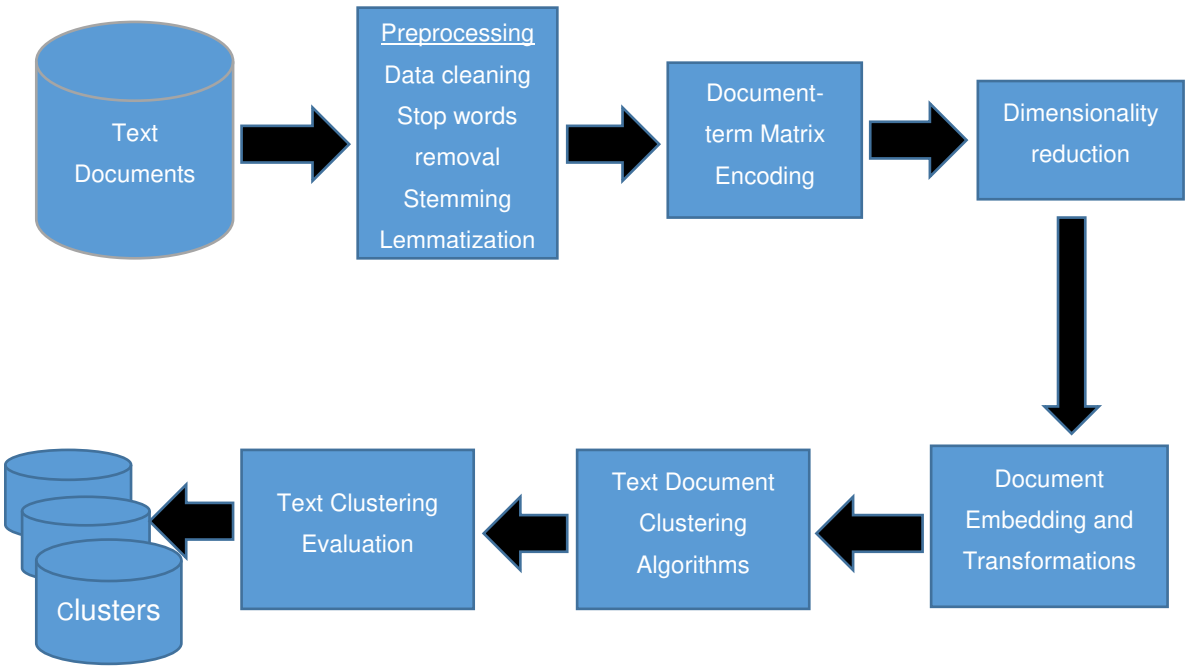


Figure 4.1. Experimental procedure for text document clustering.

The k-means and HDBSCAN clustering algorithms are used to cluster texts. Various levels of accuracy are measured by evaluating the resulting clusters using text clustering evaluation tools. To obtain the best accuracy and the most clusters, hyper parameter tuning and relative cluster validation were also used. When utilizing deep learning models for clustering, there are a few changes to these procedures. The concrete steps involved in performing text document clustering using machine learning and python as a programming language are listed below:

1. Take data from the data source and read it.
2. Import auxiliary functions and libraries from Python.
3. Perform exploratory data analysis on the data.
4. Clean and preprocess the data.
5. Using NLP, convert the document-term matrix from strings to numbers.
6. Perform dimensionality reduction on the data.
7. Document embeddings and transformations using sentence transformers and UMAP.
8. Text document clustering using HDBSCAN and k-means algorithms
9. Text document clustering visualization and evaluation
10. Relative validation and hyper-parameter Tuning.

The concrete steps required for text document clustering and deep clustering algorithms in python are listed below:

1. Take data from the data source and read it.
2. Import auxiliary functions and libraries from Python.
3. Perform exploratory data analysis on the data.
4. Clean and preprocess the data.
5. Using NLP, convert the document-term matrix from strings to numbers.
6. Perform dimensionality reduction on the data.
7. Split the data into train and test sets.
8. Define parameters for the deep learning architecture.
9. Train the deep learning model
10. Perform model prediction
11. Perform sentence embedding and reduce dimensions using UMAP.
12. Perform the model clustering task using HDBSCAN and k-means
13. Visualize the clustered model and evaluate the resultant clusters
14. Tune hyper parameters to achieve optimal performance

4.2.2. Experimental Procedure for Text Document Topic modeling

Probabilistic document topic models are graphical models that can be used to summarize a large collection of documents with a small distribution of all words. These distributions are

called "topics". Topic modeling denotes the task of identifying topics that best represent a set of documents. Simply put, topic modeling is the process of taking a set of unstructured and unlabeled textual data and classifying it into different themes that represent it. The main is to capture the main themes running throughout the text document collection when adapted to the text document data. This is important for understanding the content of text document clusters created by text clustering algorithms in the previous experiment. The figure 4.2 infographic describes the steps involved in topic modeling of text documents

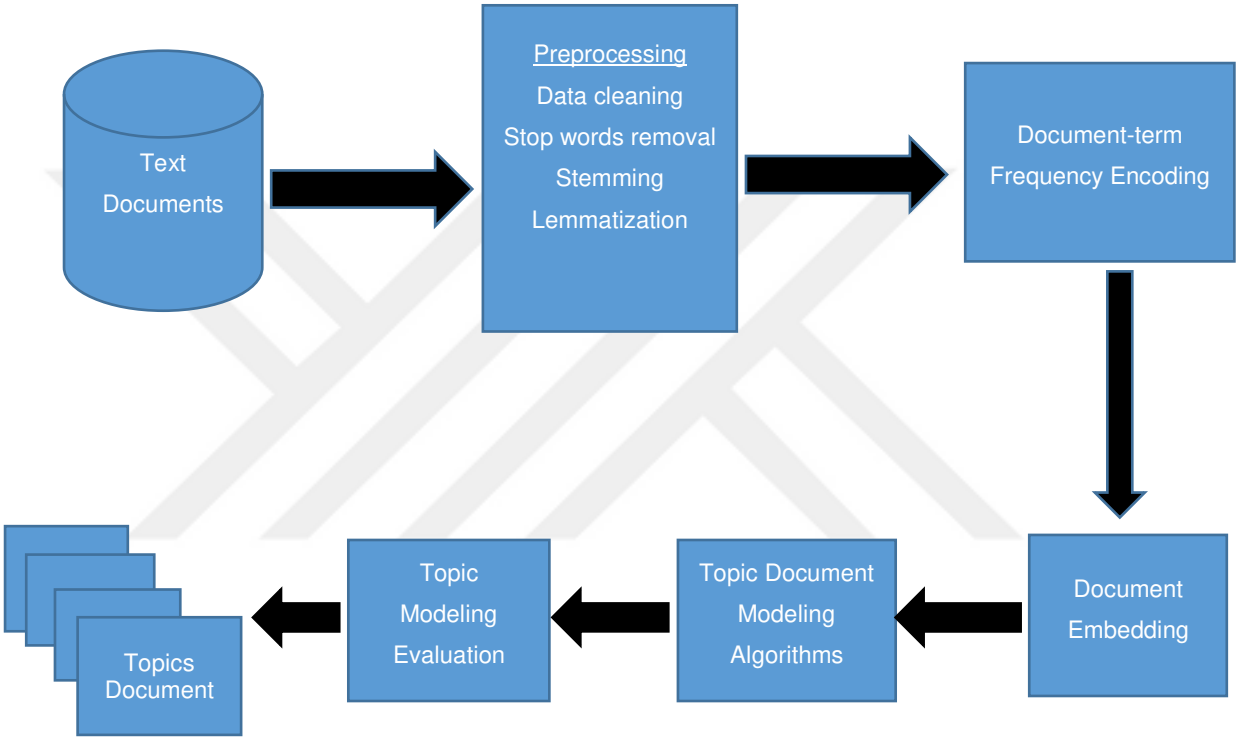


Figure 4.2. Experimental procedure for text document topic modeling

The steps for text document topic modeling are almost identical to the text document clustering algorithm except for dimensionality reduction and substitution of text document clustering algorithms with topic modeling algorithms. Three topic document-modeling algorithms were used for topic modeling of coronavirus vaccine tweets namely Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), Latent Dirichlet Allocation (LDA), and topicBERT.

4.2.2.1 Latent Dirichlet Allocation (LDA)

LDA displays documents as a mix of topics. Likewise, a topic is a collection of words. There are two parts to LDA:

1. Words that are recognized as being part of a document.

2. It is necessary to determine the words that are related to a topic or the likelihood that certain words are related to a topic.

The following describes the algorithm used to carry out the second part of LDA:

- Each word in each document is randomly assigned by the LDA model to one of the k topics (k is predetermined).
- Analyze each word w in each document d and calculate:
  - I. ***P* (topic t | document d):** the percentage of words in document d that are classified as belonging to topic t. This makes an attempt to quantify the number of words that are specific to a certain topic (t) in a given document (d). excluding the word that is being used right now. It is more likely that the word w belongs to t if several words from d do.
    (#words in d with t +alpha/ #words in d with any topic+ k*alpha)
  - II. ***P* (word w| topic t):** the percentage of tasks related to topic t over all documents originating from this term w. Due to the word w, this tries to determine how many texts exist in topic t.

Therefore, even if w is added to t, not many of these documents will be added. Updating the likelihood that the term w belongs to subject t:

P (word w with topic t) = ***P*** (topic t | document d) * ***P*** (word w | topic t). (4.1)

A word's likelihood of appearing in a topic determines how strongly all documents containing that word w will also be associated with that topic t. Similar to this, if w is not very likely to be in t, the documents that contain w will have a very low likelihood of being in t since the other words in d will be related to different topics, therefore d will have a greater probability for those topics.

The steps taken for LDA text document topic modeling in this research using Python programming language are:

1. Read in the text data from the source
2. İmport libraries and helper functions
3. Exploratory data analysis
4. Data preprocessing
5. Build N-gram models (Bigram and Trigram)
6. Create term-document frequency
7. Train the model using the LDA algorithm
8. Visualize keywords in topics
9. Evaluate the model
10. Hyper-parameter tuning

4.2.2.2 Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM)

The GSDMM is similar in many ways to the LDA, but there are remarkable differences between them. GSDMM is specifically intended to detect topics within small documents, assuming only one topic per document. This makes it suitable for topic modeling of microblogging content such as COVID-19 vaccine tweets. Another difference between the two approaches is that in the case of LDA, the number of topics needs to be specified in advance, whereas GSDMM requires only the upper limit (maximum number of topics), and from there, the number of topics in the data is inferred automatically. The steps taken to perform GSDMM topic modeling in this research are listed below:

1. Take data from the data source and read it.
2. Import auxiliary functions and libraries from Python.
3. Perform exploratory data analysis on the data.
4. Clean and preprocess the data.
5. Build N-gram models(bigram and trigram).
6. Creation of bag of words dictionary.
7. Train the model using the GSDMM algorithm.
8. Visualize keywords in topics
9. Evaluate the model
10. Perform hyperparameter tuning

4.2.2.3 BERTopic / TopicBERT

In BERTopic / TopicBERT topic modeling, representations from transformers are used to obtain quality word embeddings, which are then passed to the BERTopic algorithm for subsequent dimensionality reduction using UMAP, clustering using HDBSCAN, and extraction of topics from the formed clusters. Topical cluster names and documents' distributions for different topics are observed.

Steps taken to perform TopicBERT in this research are listed below:

1. Take data from the data source and read it.
2. Import auxiliary functions and libraries from Python.
3. Perform exploratory data analysis on the data.
4. Clean and preprocess the data.
5. Create text document embeddings
6. Train the model using the TopicBERT algorithm.
7. Visualize topics representation
8. Perform hyperparameter tuning.

5. RESULTS AND FINDINGS

The research findings and results of the study are presented in this chapter. The research's coronavirus vaccination data sample was provided. A presentation of exploratory data analysis was also made in order to comprehend the descriptive statistics used in the data. Using unsupervised learning techniques (machine learning and deep learning algorithms), clusters were created from the available tweets on the coronavirus vaccine. GSDMM, LDA, and TopicBert topic modeling techniques were used to show the coronavirus vaccine corpus's important subjects and phrases. These unsupervised algorithms' performance was evaluated using a variety of assessment criteria, including the silhouette score, Davies-Bouldin index, and Calinski-Harabasz index. This study proved that word embeddings are the most effective way to create word vectors for clustering applications. The coronavirus vaccination data was assigned clusters, which may be utilized for supervised learning tasks like classification and pattern recognition.

5.1. Exploratory Data Analysis

Figure 5.1 showed the dataset features of the COVID-19 vaccine data used for this study in a tabular format. A presentation of the said data stored in Comma Separated Values (CSV) is shown below. The dataset was transformed to a tabular data format using the pandas Data frame library.

	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date	text	hashtags
0	MyNewsNE	Assam	MyNewsNE a dedicated multi-lingual media house from North Eastern India.	24-05-2020 10:18	64.0	11.0	110.0	False	18-08-2020 12:55	Australia to Manufacture Covid-19 Vaccine and give it to the Citizens for free of cost: AFP quotes Prime Minister\n#CovidVaccine	['CovidVaccine']
1	Shubham Gupta	NaN	I will tell about all experiences of my life from my videos hope that you all like the videos 🙂	14-08-2020 16:42	1.0	17.0	0.0	False	18-08-2020 12:55	#CoronavirusVaccine #CoronaVaccine #CovidVaccine Australia is doing very good https://t.co/kBT7l6pArY	['CoronavirusVaccine', 'CoronaVaccine', 'CovidVaccine']
2	Journal of Infectiology	NaN	Journal of Infectiology (ISSN 2689-9981) is accepting submissions for an upcoming Volume 3 Issue 2. For further queries contact editor@infectiologyjournal.com.	14-12-2017 07:07	143.0	566.0	8.0	False	18-08-2020 12:46	Deaths due to COVID-19 in Affected Countries\nRead More: https://t.co/V8Y3Stu0UW\n@r_piryani @shitalbhandary... https://t.co/6jpMxX2KtI	NaN
3	Zane	NaN	Fresher than you.	18-09-2019 11:01	29.0	25.0	620.0	False	18-08-2020 12:45	@Team_Subhashree @subhashreesotwe @iamrajchoco Stay safe @subhashreesotwe di & @iamrajchoco da ❤️❤️... https://t.co/ayhoaQm0Ls	NaN
4	Ann-Maree O'Connor	Adelaide, South Australia	Retired university administrator. Melburnian by birth, now living in Adelaide. Look back fondly to the Whitlam years; one of Keating's true believers.	24-01-2013 14:53	83.0	497.0	10737.0	False	18-08-2020 12:45	@michellegrattan @ConversationEDU This is what passes for leadership in our country: a voucher for something that w... https://t.co/OUUb1PeYIj	NaN

Figure 5.1. A sample COVID-19 vaccine tweet in pandas dataframe.

The features of the dataset which constitutes the columns are shown below in figure 5.2. It can be seen that there are 368,588 rows of data within each column.

```
[ ]  # list the number of columns or features
     print(df.columns)

     Index(['user_name', 'user_location', 'user_description', 'user_created',
            'user_followers', 'user_friends', 'user_favourites', 'user_verified',
            'date', 'text', 'hashtags', 'source', 'is_retweet'],
           dtype='object')

[ ]  # list the shape of the dataframe rows or values
     print(df.shape)

     (368588, 13)

     #get the information of respective columns
     df.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 368588 entries, 0 to 368587
     Data columns (total 13 columns):
      #   Column            Non-Null Count   Dtype
     ---  ------            --------------   -----
      0   user_name         368574 non-null  object
      1   user_location     289065 non-null  object
      2   user_description  348130 non-null  object
      3   user_created      368563 non-null  object
      4   user_followers    368549 non-null  float64
      5   user_friends      368549 non-null  object
      6   user_favourites   368549 non-null  object
      7   user_verified     368549 non-null  object
      8   date              368547 non-null  object
      9   text              368549 non-null  object
      10  hashtags          306913 non-null  object
      11  source            366135 non-null  object
      12  is_retweet        368526 non-null  object
     dtypes: float64(1), object(12)
     memory usage: 36.6+ MB
```

Figure 5.2. COVID-19 vaccine tweet dataset features in Pandas Dataframe.

From the data frame information, it can be seen that some of the data columns have missing values. Therefore, figure 5.3 below explained the percentage of missing values for each particular attribute or feature in the dataset. It can be observed that the user location feature has the highest percentage of missing values with more than 20 percent of its entries having missing values, followed by hashtags with 17 percent, user description with 6 percent, and source of the tweets which explains where the tweets originate with 1 percent. It is however worthy to note that the text column, which contains the COVID-19 vaccine tweets that this research is keenly interested in, has about 39 missing values, which is almost negligible in percentages. The reason for the missing values is because user location and user description are optional fields when creating a tweeter account and also a user can tweet without using hashtags as well as it is optional too.

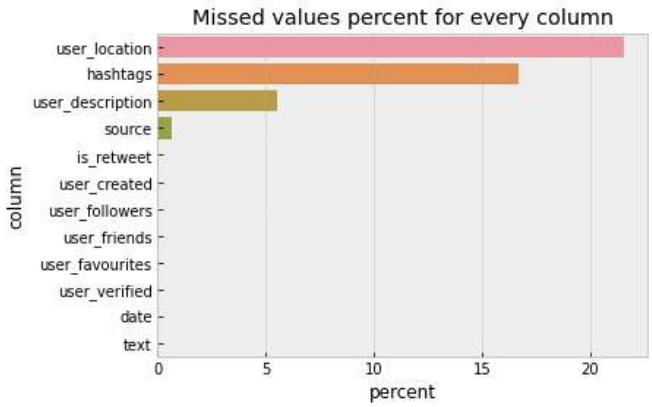


Figure 5.3. Percentage of Missing Values in the COVID-19 Vaccine Tweet Dataset Features.

Figure 5.4 below presented the percentage of unique values for each COVID-19 vaccine dataset feature. It can be seen from the data that about 25 percent of user locations have missing values, 17 percent of hashtags have missing values, 6 percent of user descriptions have missing values, and about 1 percent of the source of tweets have missing values.

	user_name	user_location	user_description	user_created	user_followers	user_friends	user_favourites	user_verified	date	text	hashtags	source	is_retweet
Total	368574.000	289065.000	348130.000	368563.000	368549.000	368549.000	368549.000	368549.000	368547.000	368549.000	306913.00	366135.000	368526.0
Uniques	152337.000	39777.000	156913.000	155807.000	27415.000	21437.000	80383.000	13.000	368908.000	368211.000	131912.00	387.000	1.0
Percent from total	41.331	13.761	45.073	42.274	7.439	5.817	21.811	0.004	97.385	99.908	42.98	0.106	0.0

Figure 5.4. Percentage of unique values in the COVID-19 vaccine tweet dataset features.

Figure 5.5 presented information on the location of individuals and organizations that are tweeting about the COVID-19 vaccine. The figure showed that most percentages of individuals and organizations tweeting are from India, followed by the United States of America, the United Kingdom, Canada, and Australia in that order.

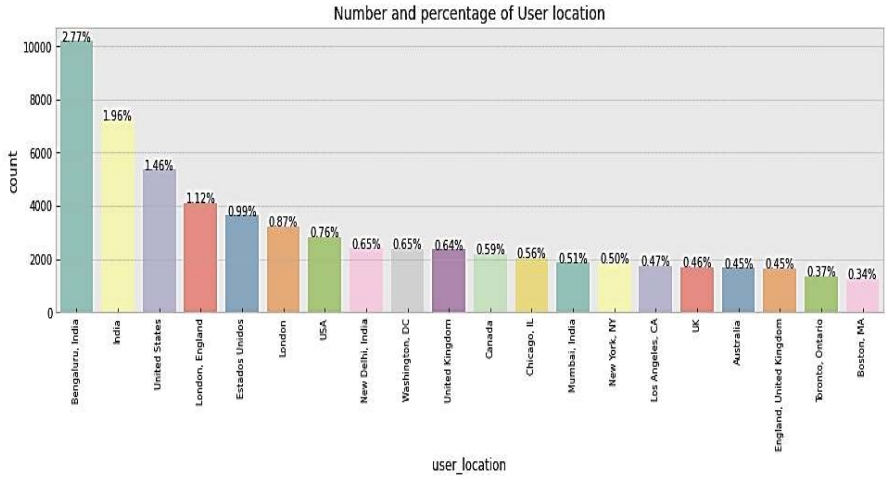


Figure 5.5. User location of individuals and organizations tweeting about the COVID-19 vaccine.

Figure 5.6 presented an infographic of the count of user names of individuals and organizations tweeting about the COVID-19 vaccine. VaxBLR, an automated hourly update on free and paid COVID-19 availability across Bengaluru India, was the most source of tweets with almost 10,000 vaccine tweets, followed by COVID News a page dedicated to worldwide news on the COVID-19 pandemic. CVS and Rite Aid Vaccine Finder (in California, Virginia, and Maryland) are next in line providing a combined 0.89 percent of the vaccine tweets. Others are India today and other individuals.

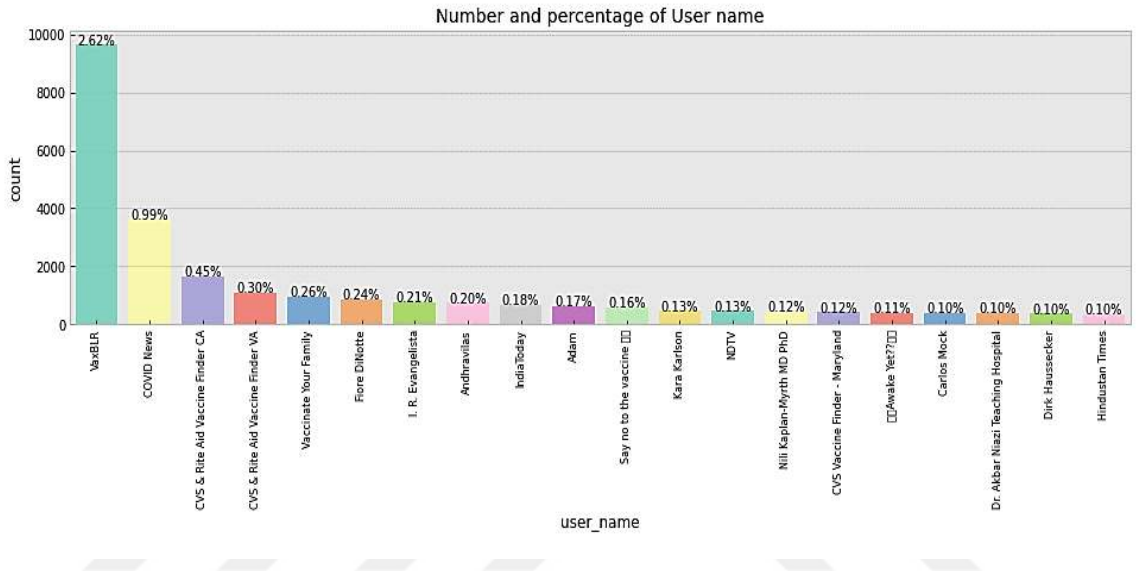


Figure 5.6. User names of individuals and organizations tweeting about the COVID-19 vaccine.

Figure 5.7 presented the count and percentages of device sources tweeting about the COVID-19 vaccine. Twitter Web App, which is the web-based version of Twitter, provided the bulk of the devices tweeting about the COVID-19 vaccine with 30.47 percent of tweets, followed by iPhone users with almost 29 percent of tweets.

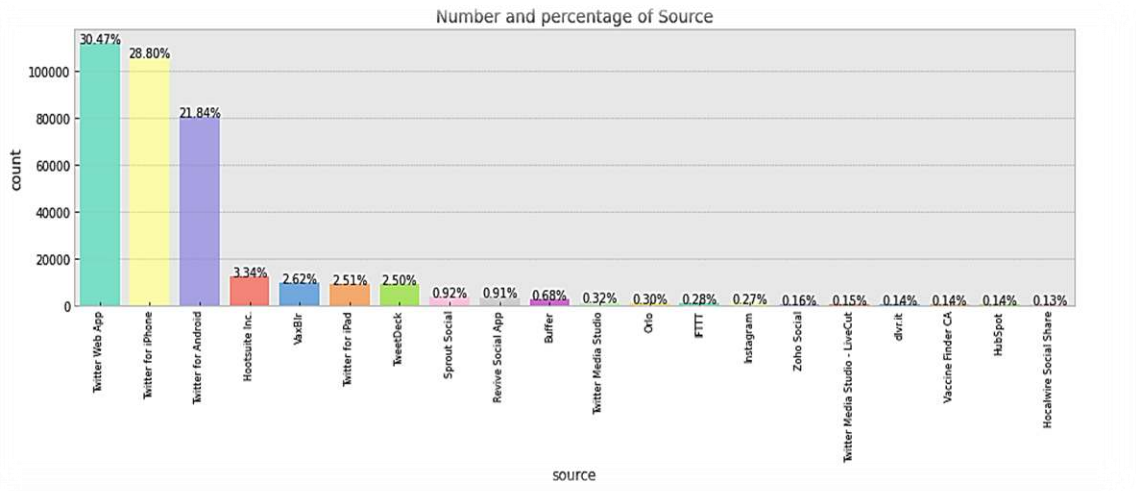


Figure 5.7. Sources of devices used by individuals and organizations tweeting about the COVID-19 vaccine.

Android users contributed 21.84 percent of the vaccine tweets. Altogether, the Twitter web app, iPhone, and Android devices contributed about 80 percent of COVID-19 vaccine tweets.

Figure 5.8 presented a world cloud of COVID-19 tweets around the whole world. Words like Dose 1, Dose 2, COVIDvaccine, COVISHIELD, Slots, and Availability dominated the COVID-19 vaccine tweets.

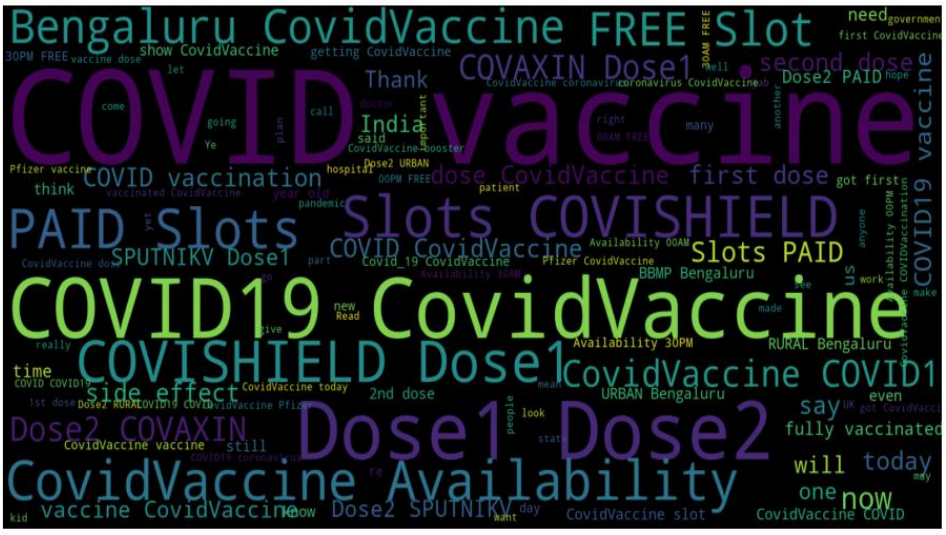


Figure 5.8. World cloud of tweets about the COVID-19 vaccine around the world.

Figure 5.9, figure 5.10, figure 5.11, and figure 5.12 presented the world clouds of tweets prevalent in the most popular regions represented by the user location information of the tweets (India, Canada, the United Kingdom, and the United States of America). Almost the same theme can be seen all over individual countries.

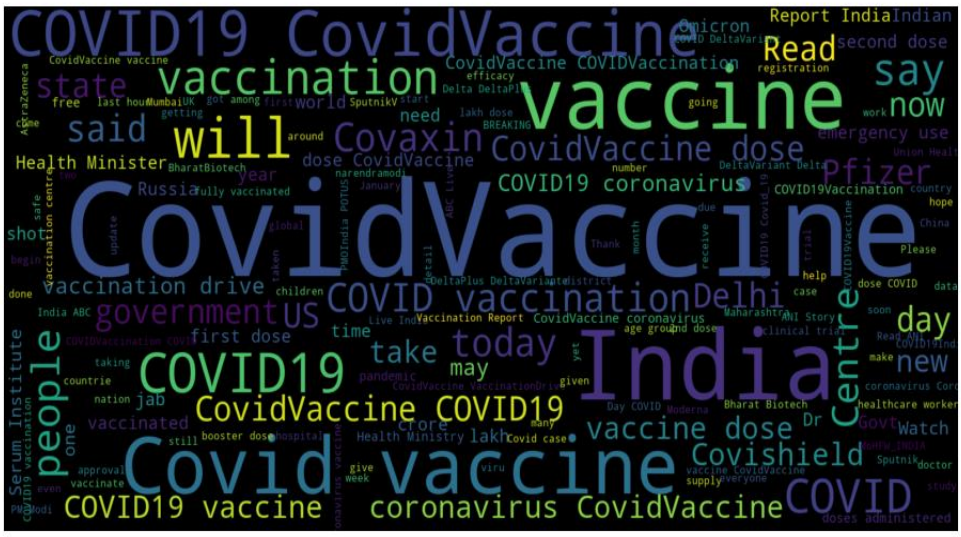


Figure 5.9. World cloud of tweets about the COVID-19 vaccine from India.

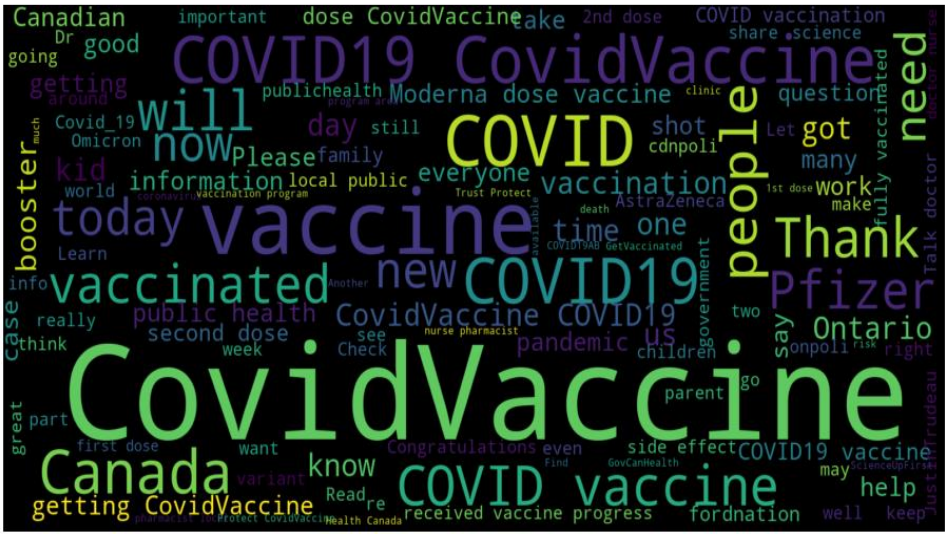


Figure 5.10. World cloud of tweets about the COVID-19 vaccine from Canada.

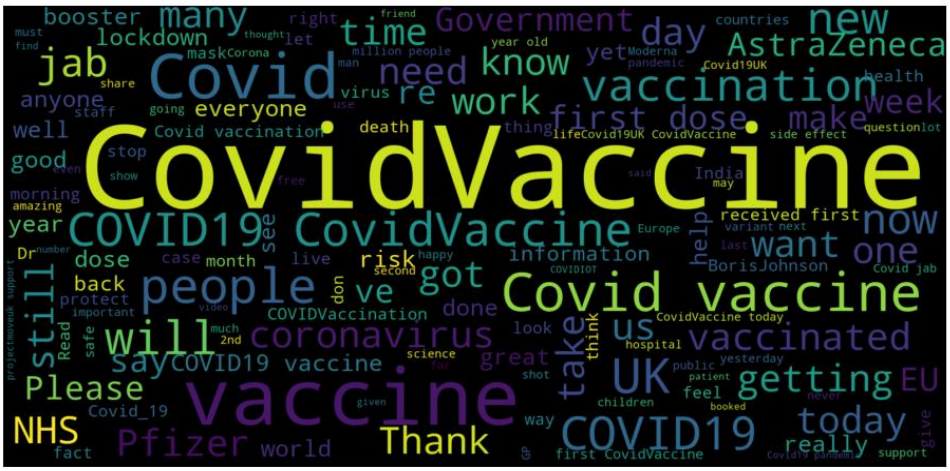


Figure 5.11. World cloud of tweets about the COVID-19 vaccine from the United Kingdom.

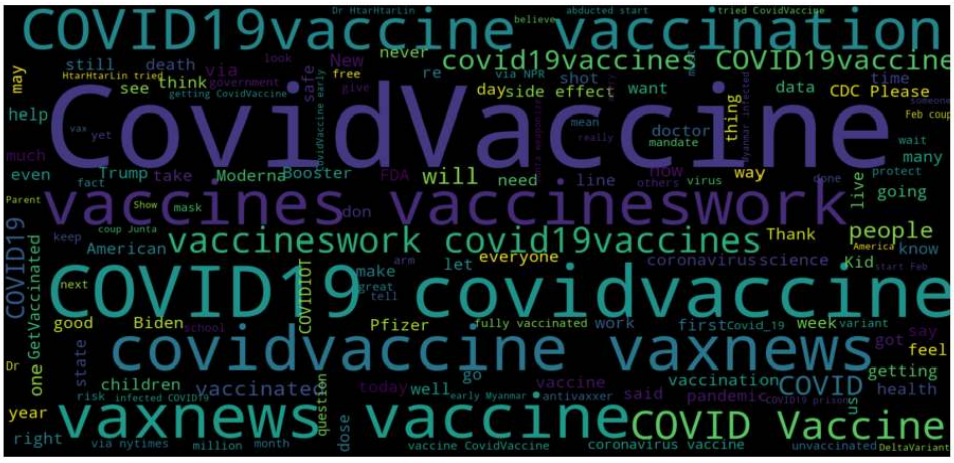


Figure 5.12. World cloud of tweets about the COVID-19 vaccine from the United States of America.

Figure 5.13 presented a density count of hashtags that are associated with the COVID-19 vaccine tweets. It can be seen that a good number of tweets have between 0 to 10 hashtags per tweet.

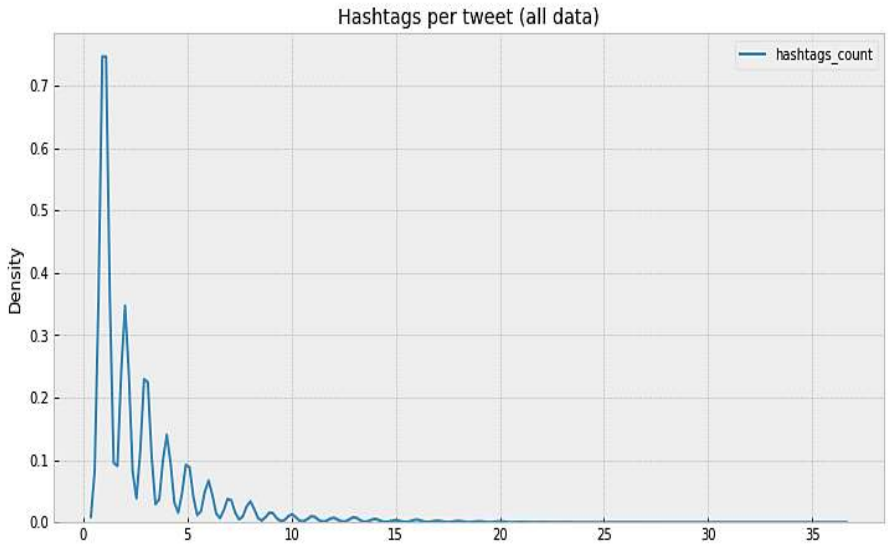


Figure 5.13. Density and count of hashtags of the COVID-19 vaccine tweets.

Figure 5.14 presented the time variation and volume of COVID-19 vaccine tweets between the time intervals under study. It can be seen that the vaccine tweets peaked around December 2020 and it is still being discussed today.

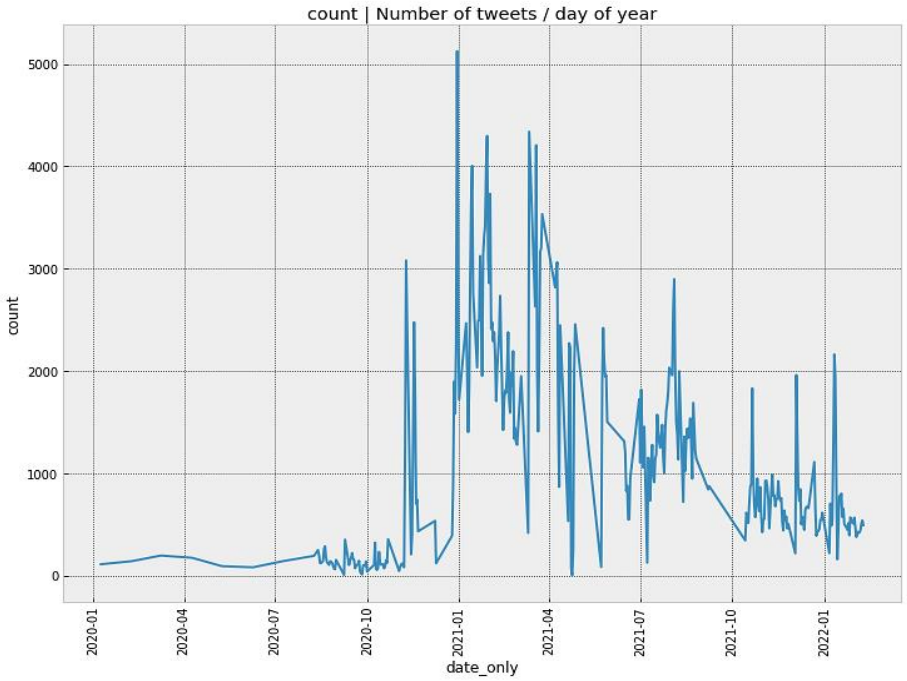


Figure 5.14. Time variations of the COVID-19 vaccine tweets.

5.2. Machine Learning-Based Clustering

In this research study, Machine learning-based data processing, dimensionality reduction, and clustering were done on the COVID-19 vaccine dataset. The two major clustering algorithms employed are k-means (a partition-based clustering algorithm) and HDBSCAN (a hybrid of hierarchical and density-based clustering algorithms). The relevant results of the clustering process are shown below. Figure 5.15 presented a result of using quality word embeddings from sentence transformers to cluster COVID-19 vaccine tweets using a k-means clustering algorithm. The diagrams also presented the features that were learned during this clustering process, which are represented as the strongest features below.

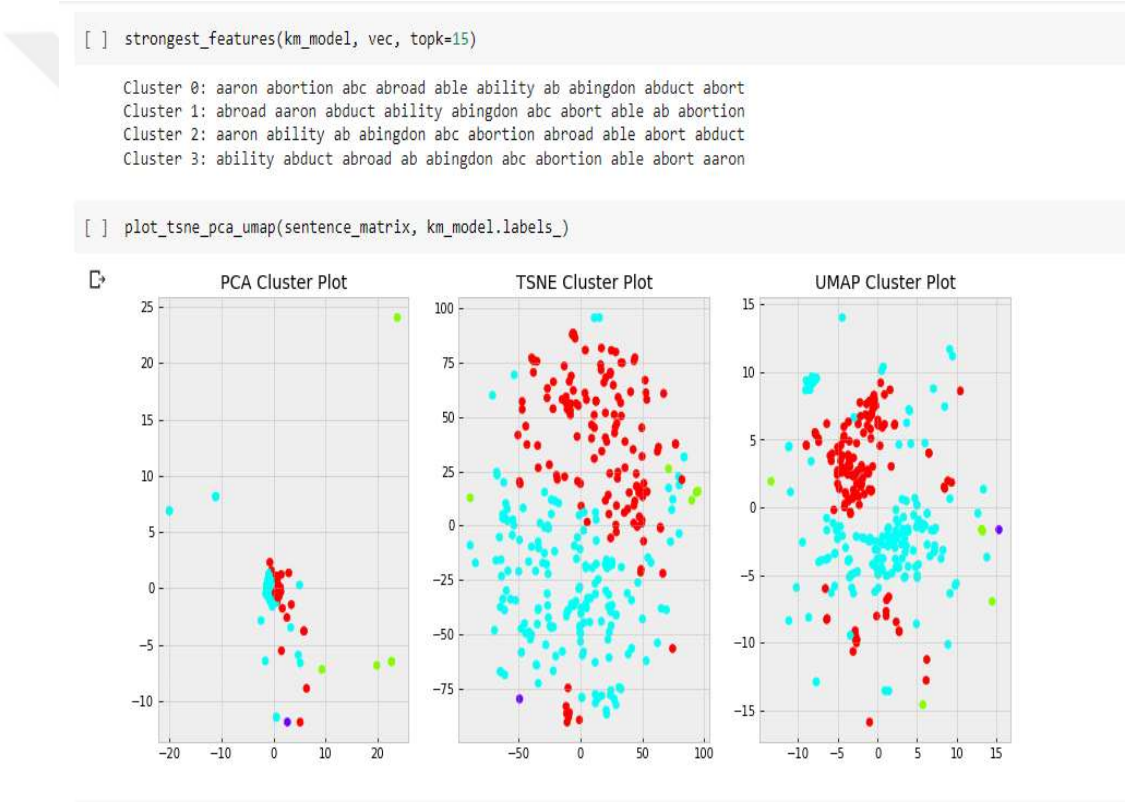


Figure 5.15. Sentence embedding and k-means clustering on the COVID-19 vaccine tweets.

Figure 5.16 presented the result of performing clustering on quality word vectors from sentence transformers using the HDBSCAN clustering algorithm. The clustering process produced six clusters. With cluster one being the most dominant cluster and the other five clusters being the smaller clusters with distinctive qualities. The strongest features of these clusters are also shown.

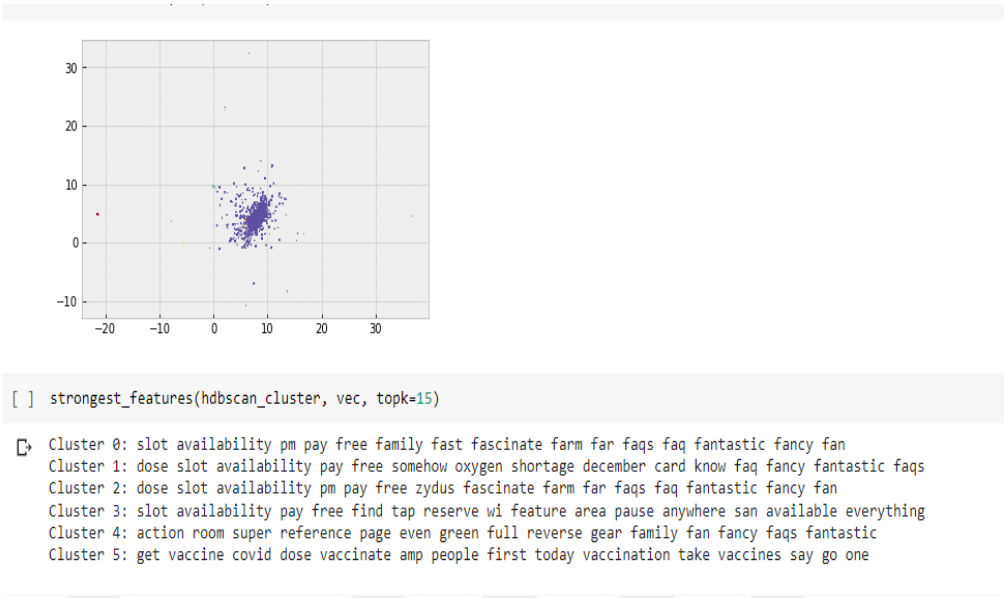


Figure 5.16. Sentence embedding and HDBSCAN clustering on the COVID-19 vaccine tweets.

5.3. Deep Learning-Based Clustering

In this research study, deep learning-based data processing, dimensionality reduction, and clustering were done on the COVID-19 vaccine dataset. The two major clustering algorithms employed are k-means (a partition-based clustering algorithm) and HDBSCAN (a hybrid of hierarchical and density-based clustering algorithms). The relevant results of the clustering process are shown below. Figure 5.17 presented a result of using quality word embeddings from sentence transformers and AutoEncoder to cluster COVID-19 vaccine tweets using a k-means clustering algorithm. The AutoEncoder-based clustering using embeddings from sentence transformer and k-means clustering algorithm produced six clusters. From figure 5.17 below, it can be seen that the dominant cluster is cluster 0, followed by clusters 1, and 2,3,4,5 in that other. Figure 5.17 also presented the features that were learned during this clustering process, which are represented as the strongest features below.

```
[ ] strongest_features(k_means2, vec, topk=15)

Cluster 0: aaron accuse abandon achieve acceptable abuse account absolutely accomplish acquire accurate activities able add abroad
Cluster 1: aaron abuse account accurate achievement absolute accelerate abroad abortion accept action accident actively ability acquire
Cluster 2: abduct activate acquire achy add abuse accelerate actually achieve account abc abandon acworth access able
Cluster 3: aaron abandon account abuse achievement actual abroad accept abortion abc accelerate absolute actively action accident
Cluster 4: abc abandon abuse account accurate achievement achieve actual accept action acquire abroad accelerate actively ability
Cluster 5: account abuse accurate achievement accept accelerate achieve abroad ability abortion acquire actual absolute action actively
```

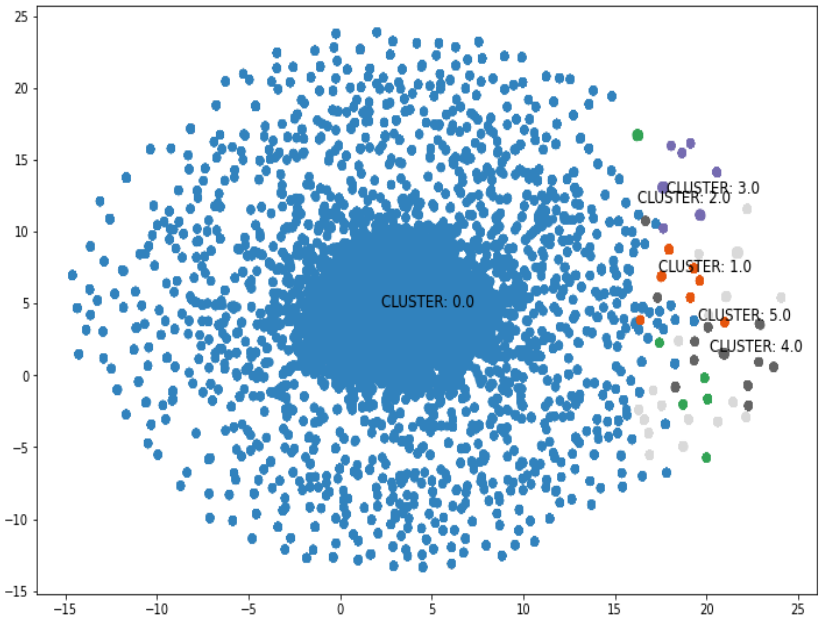
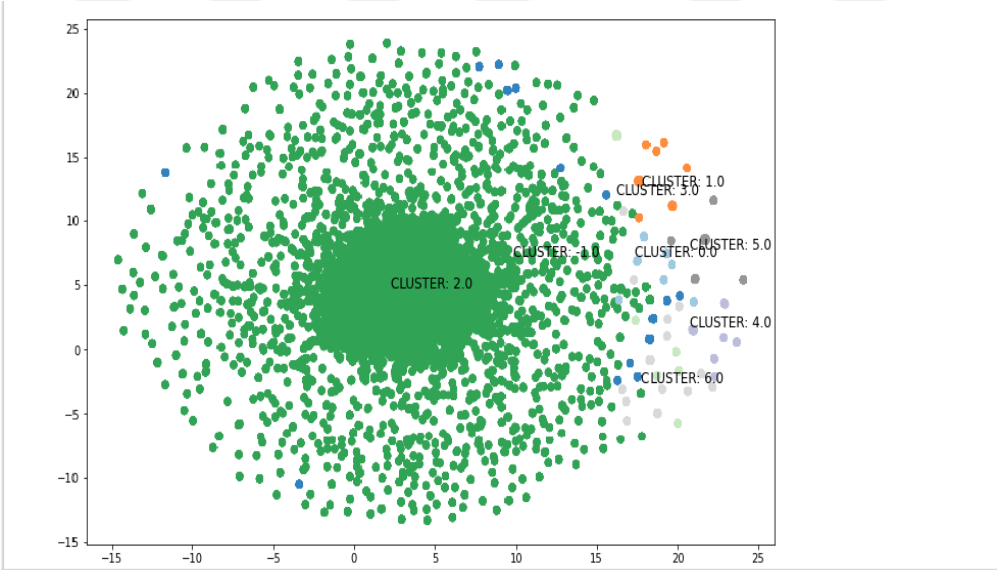


Figure 5.17. AE sentence embedding and k-means clustering on the COVID-19 vaccine tweets.

Figure 5.18 below, presented the result of performing clustering on quality word vectors from sentence embeddings using the HDBSCAN clustering algorithm.




```
[ ]  strongest_features(hdbscan_cluster, vec, topk=15)

Cluster 0: dose slot availability pay free zydus fair famous family families familiar fam false fall fake
Cluster 1: slot availability pay free failures fan famous family families familiar fam false fall fake faith
Cluster 2: get vaccine covid dose vaccinate amp people first today vaccination take vaccines say one need
Cluster 3: dose slot availability pay free zydus fair famous family families familiar fam false fall fake
Cluster 4: dose slot availability pay free zydus fair famous family families familiar fam false fall fake
Cluster 5: dose slot availability pay free zydus fair famous family families familiar fam false fall fake
Cluster 6: dose slot availability pay free zydus fair famous family families familiar fam false fall fake
```

Figure 5.18. AE sentence embedding and HDBSCAN Clustering on the COVID-19 vaccine tweets.

The AutoEncoder-based clustering using embeddings from sentence transformer and HDBSCAN clustering algorithm produced seven clusters. From figure 5.18 above, it can be seen that the dominant cluster is cluster 2, followed by clusters 1, 6, 3, 4, 5, and 0 in that other. Figure 5.18 also presented the features that were learned during the clustering process, represented as the strongest features above.

5.4. Topic Modeling

COVID-19 vaccine tweets topic modeling was done using two machine learning-based topic-modeling algorithms, which are Latent Dirichlet Allocation (LDA) and Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), as well as a deep learning-based topic modeling algorithm TopicBERT / BERTopic (a topic modeling based on Bidirectional Encoder Representations from Transformers (BERT). These algorithms differ in the way and manner they extract latent topics from the COVID-19 vaccine corpus.

5.4.1. Topic Modeling using LDA

Figure 5.19 presented the result of topic modeling with LDA. The figure presented the inter-topic distance between the various topics as well as the top thirty most salient terms in each topic. The figure is an interactive plot showing the overall term frequency of the whole COVID-19 vaccine tweets and the estimated term frequency within the selected topics.

The overall salient terms in the document are highlighted in a light blue color while individual topic salient terms are highlighted in red color. The estimated salient terms frequency in the next topic can be viewed by simply pressing the next topic button while the estimated salient term for the previous topic can be viewed by pressing the previous topic button on the plot. The salient terms for a particular topic can as well be viewed by entering the topic number in the selected topic column. Overall, there are 302 topics generated for the entire document in this plot.

The size of the circles shows the marginal topic distribution, which can be 2 percent, 5 percent, or 10 percent as shown below. Other important metrics such as the percentage contribution of individual scores to the topic, word count for each dominant topic, coherence score, perplexity, and so on can be seen in appendix-1.

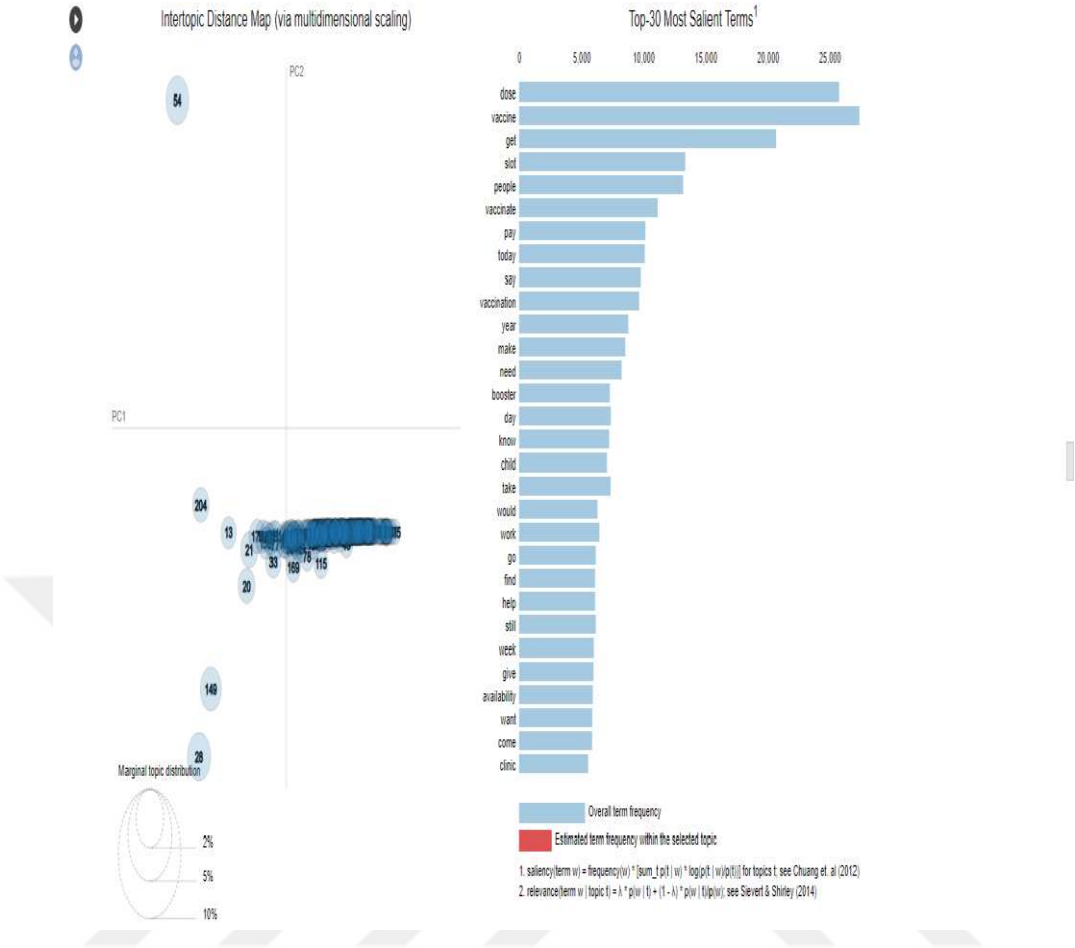


Figure 5.19. LDA-based topic modeling on the COVID-19 vaccine tweets.

Figure 5.20 below presented the dominant topics for the LDA-based topic model. The data frame consists of the document number, dominant topic, percentage of contribution to the topic, keyword in the specific topic, and the cleaned text associated with the topic.

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	202.0	0.0172	refuse, treatment, reason, cover, cost, insurance, company, deny, hospital, regime	australia manufacture covid vaccine and give the citizens for free cost afp quotes prime minister
1	1	0.0	0.0033	arrive, deliver, batch, handle, consignment, box, flight, carry, movement, shipment	australia doing very good
2	2	176.0	0.0195	affect, woman, impact, period, evidence, fertility, explain, cycle, man, change	deaths due covid affected countries read more
3	3	175.0	0.0229	stay, home, tune, nee, forever, bind, order, leave, lock, date	stay safe amp
4	4	285.0	0.0216	pass, test, place, therapy, gene, vaccine, benefit, product, vector, sample	this what passes for leadership our country voucher for something that
5	5	104.0	0.0195	risk, benefit, increase, factor, put, associate, outweigh, complication, harm, pose	the multi system inflammatory syndrome children mis atypical kawasaki disease the
6	6	0.0	0.0033	arrive, deliver, batch, handle, consignment, box, flight, carry, movement, shipment	
7	7	134.0	0.0151	kid, parent, child, rush, meet, watch, share, include, trial, superhero	well let qualify that would anyone any party get vaccine rushed out and minimally tested coming from russia
8	8	193.0	0.0232	country, world, develop, globally, patent, access, produce, recognise, property, block	most countries without the ability make locally will forced rely others like the china
9	9	238.0	0.0178	hear, view, opinion, voice, reason, debate, argument, speak, express, viewer	zooms charts week hear episode

Figure 5.20. Dominant Topics of the LDA-based Topic Modeling on the COVID-19 Vaccine Tweets.

Figure 5.21 presented the top five sentences for the whole COVID-19 vaccine tweets. The data frame consists of the topic number under consideration, the topic percentage contribution, the keywords prevalent in the topic, and the cleaned text stating the sentence.

	Topic_Num	Topic_Perc_Contrib	Keywords	Text
0	0.0	0.1693	arrive, deliver, batch, handle, consignment, box, flight, carry, movement, shipment	the sustained efforts ensure smooth delivery medical essentials airports are applaudable today shipment boxes arrived from was promptly delivered the consignee
1	1.0	0.0867	vaccine, figure, give, today, update, refer, kindly, back, minister, graph	today vaccine update for anyone who wants refer back previous days figures are above
2	2.0	0.1658	roll, vote, plan, poll, ahead, sleeve, voter, election, party, pass	new jersey you live amp are against and have freedom you must vote murphy out his admin leaked his agenda planning implementing harsh and agenda post election vote murphy out vote murphy out
3	3.0	0.1050	control, government, power, create, push, knowledge, attempt, fear, resist, abuse	evil controls our narratives via our media and institutions attempts override our common sense and logic and uses fear succeed none are alone unite with others have discussion use cash not digital
4	4.0	0.0668	people, vaccinate, urge, asap, motivate, youngster, caution, strongly, successfully, parliament	india have successfully vaccinated over million people paving the way for safer and secure nation but this only the beginning and have long way urge you all come forward and get your covid vaccine

Figure 5.21. Top 5 sentences of the LDA-based topic modeling on the COVID-19 vaccine tweets.

5.4.2. Topic Modeling using GSDMM

Figure 5.22 shows the count of topics in each GSDMM-based topic modeling of COVID-19 vaccine tweets. Topic 13 has the most number of topics with about 280,000 words, followed by topic 17 with 250,000 words, topics 20, 11, 18, 7, and 9 have about 200,000 words or more, followed by other topics with topic 15 and topic 3 having the least count of 150,000 or less.

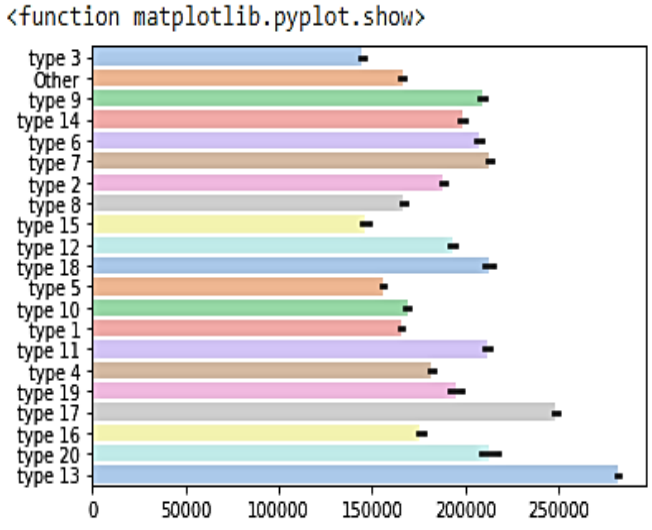


Figure 5.22. Topic counts of the GSDMM-based topic modeling on the COVID-19 vaccine tweets.

Figure 5.23 shows the percentage distribution of topics in percentages of the COVID-19 vaccine tweets. Topics 6 and topics 7 have 5.41 and 5.12 percent respectively. Topics 1 and 2 have the highest percentage with 11.5 percent and 10.5 percent respectively. Topics 5 and 8 have the least percentage figures with 0.318 percent and 1.15 percent respectively.

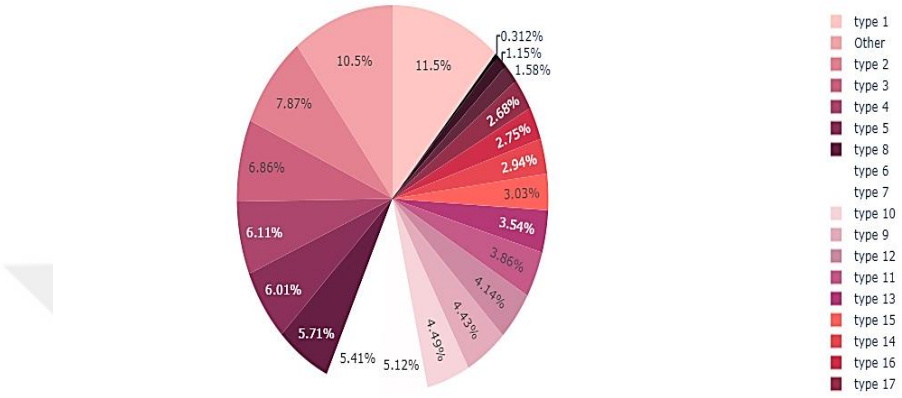


Figure 5.23. Topic distribution of the GSDMM-based topic modeling on the COVID-19 vaccine tweets.

Figure 5.24 shows the world cloud for the respective 21 topics found by the GSDMM-based topic-modeling algorithm.

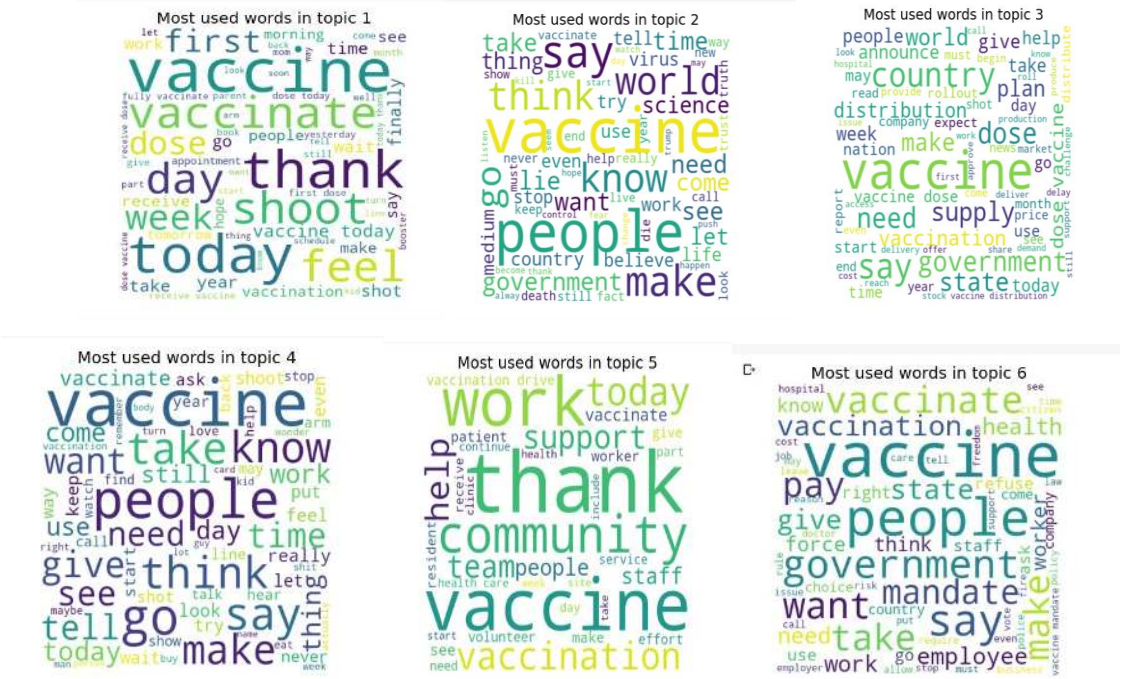


Figure 5.24. World cloud for topics in GSDMM-based topic modeling of the COVID-19 vaccine tweets.

Topic 1 have keywords such as vaccine, vaccinate, today, etc. Topics 2 have keywords such as people, vaccine, think, make, say, world, etc. Topic 3 has keywords vaccine, need, supply, government, state, country, etc. Topic 4 keywords are vaccine, take, know, people, want, give, etc. Topic 5 keywords are work, today, support, thank, help community, team, etc. Topic 6 keywords are vaccine, vaccinate, people, state, mandate, etc. topic 7 have keywords such as side effect, death, take, people, die, risk, vaccine, and patient. Topic 8 keywords are check, share, people, video, question, join, watch, information, vaccine, and talk. Topic 9 keywords are vaccine, study, immunity, infection, and dose. Appointment, today, call, people, vaccination are topic 10 keywords. Topic 11 keywords are vaccine, year, parent, vaccinate, child, kid, school, and teacher. Topic 12 are fully vaccinate, vaccine progress, and nearly administer dose. Topic 13 are dose, slot, pay, and availability. Topic 14 are wear mask, people, vaccinate, need, fully vaccinated. Topic 15 are phase trial, vaccine test, study, efficacy, scientist, research, and approve. Topic 16 are side effects, symptom, fever, pains, chills, reaction, injection site, arm, and hurt. Topic 17 keywords are vaccination, clinic, walk, appointment. Topic 19 keywords are approval, approve, vaccine, emergency use, and use authorization. The main words in topic 18 are vaccine, test, travel, state, people, require, and school. Topic 20 theme words are world, distribute million, starve, illness, rest, and save.

5.4.3. Topic Modeling using TopicBERT / BERTopic

Figure 5.25 below presented the general information on the topics found by the TopicBert model. Seventy topics and their respective counts are well described in the figure. It is worthy to note that topic -1 is not included in topic counts as it is regarded as noise that is inherent in the data. Topic 0 has the highest number of words while topic 69 has the lowest number of words.

	Topic	Count	Name
0	-1	226060	-1_the_and_be_have
1	0	109770	0_the_and_get_vaccine
2	1	10115	1_dose dose_dose_slot_slot dose
3	2	2970	2_show slot_show_slot_locations show
4	3	1976	3_russia_sputnik_russian_putin
...	...	...	...
65	64	154	64_book invite_invite phone_via book_available jan
66	65	154	65_cbd_oil_cbd oil_gummies
67	66	154	66_trial_trials_the trials_the trial
68	67	151	67_lie_more lie_truth_liar
69	68	150	68_true_truth_true true_this true

70 rows × 3 columns

Figure 5.25. Topic count for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

Figure 5.26 presented the topics discussed within a particular time, as well as the frequency of the topics discussed based on the count of the tweets for the BERTOPIC / TopicBERT model. Figure 5.27 shows the hierarchies of the topics.

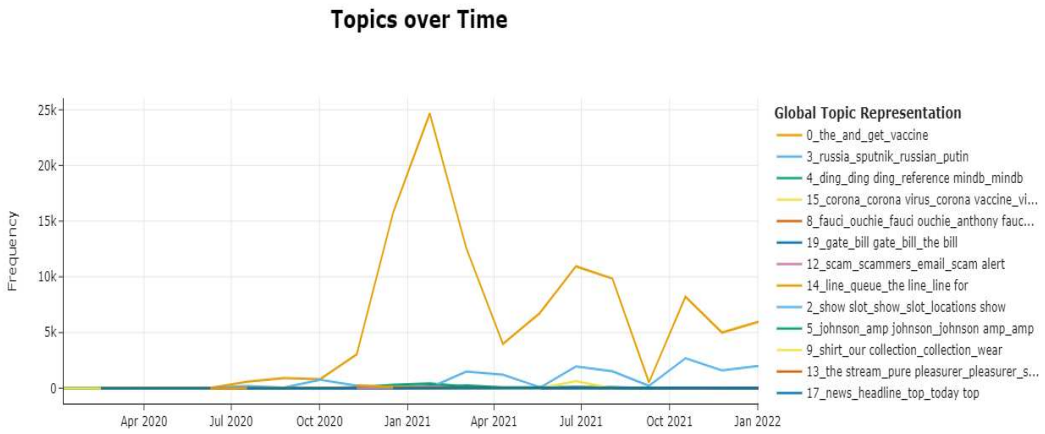


Figure 5.26. Topic distribution over time for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

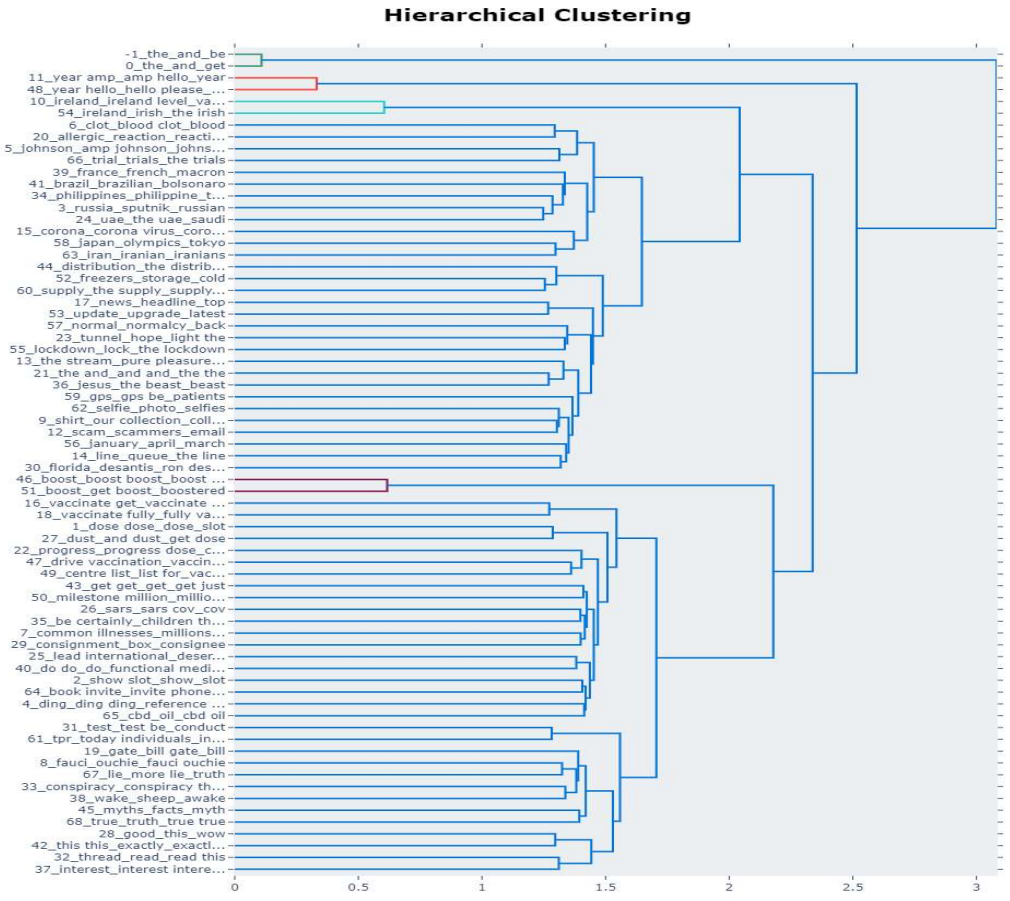


Figure 5.27. Topic hierarchies for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

Figure 5.28 presented similarity metrics that describe how similar one topic is to another topic.

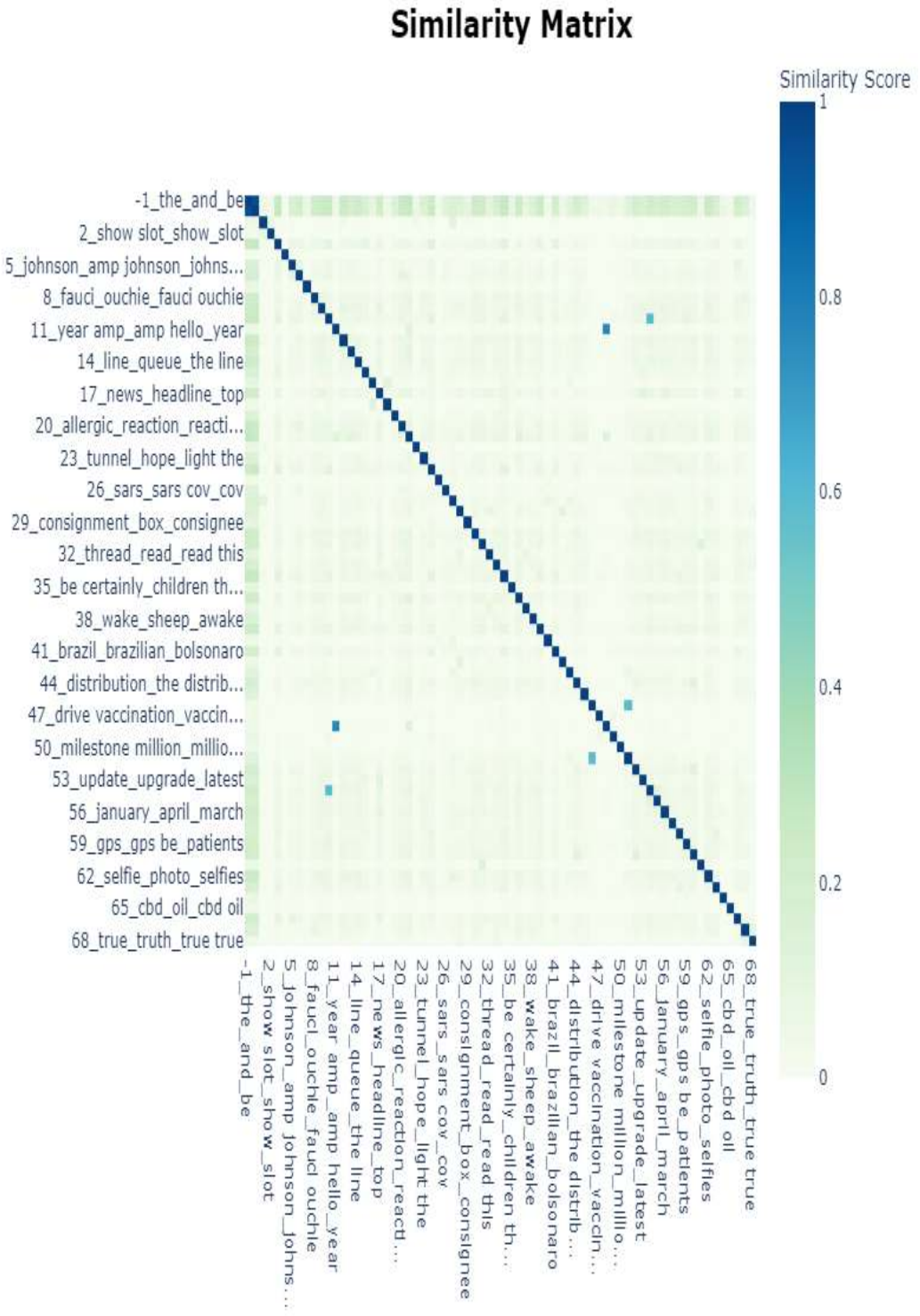


Figure 5.28. Topics similarity metrics for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

Figure 5.29 presents the words with the highest probability in each topic.

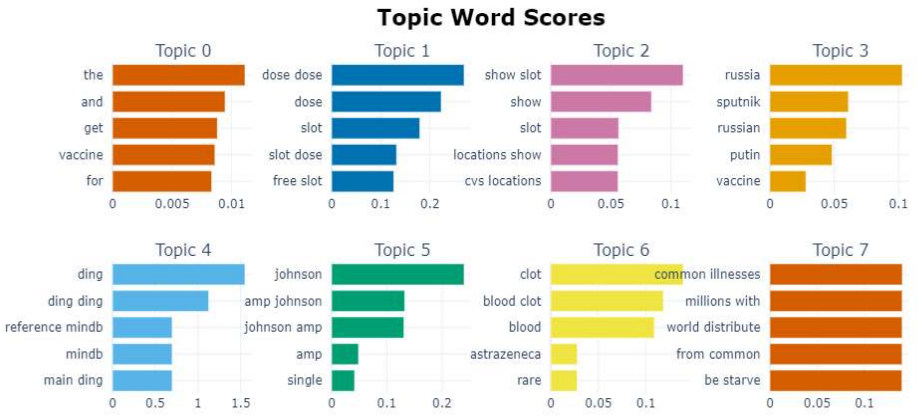


Figure 5.29. Topic word scores for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

Figure 5.30 presents how to find out the topic a particular word belongs to in the corpus.

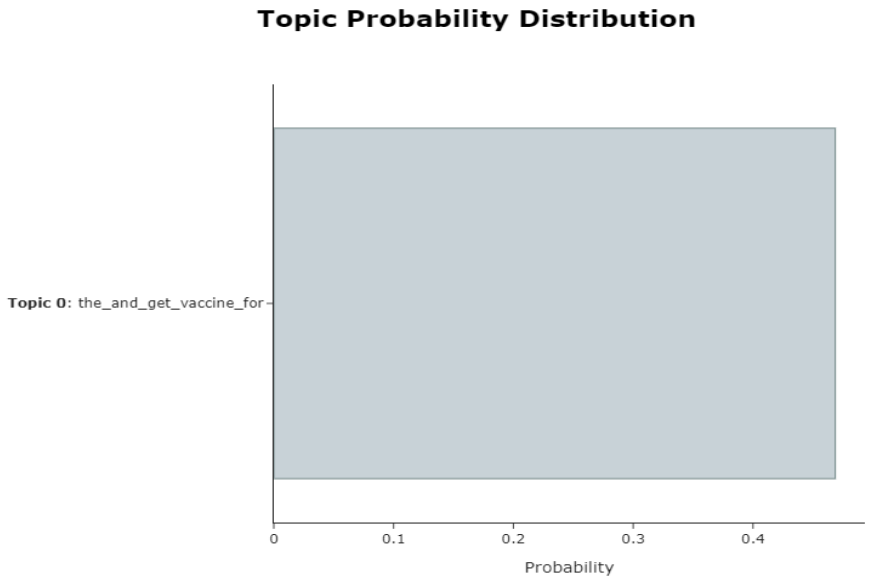


Figure 5.30. Word 20 probability distribution for TopicBERT-based topic modeling of COVID-19 vaccine tweets.

For this visualization, the word at index 20 belongs to topic 0 with a probability of more than 0.4. Table 5.1 and table 5.2 presented a summary of the results for the model evaluation of the machine learning and deep-learning based models respectively. Table 5.3 shows the summary of the model evaluation results for the topic modeling task. Table 5.4 shows the result of using TF-IDF for performing text clustering as opposed to the use of sentence transformer. Figure 5.31 presents the dataset with the assigned labels which can be used for classification task.

5.5. Model Evaluation

Table 5.1. Machine learning-based clustering with sentence transformer embedding

Evaluation Methods	Clustering Algorithms	
	K-Means	HDBSCAN
Silhouette score	0.907	0.848
Calinski Harabasz score	52,406.099	31,719.220
Davies Bouldin score	1.23	1.948

Table 5.2. Deep learning-based clustering with sentence transformer embedding

Evaluation Methods	Clustering Algorithms	
	K-Means	HDBSCAN
Silhouette score	0.960	0.885
Calinski Harabasz score	103,539.099	93,682.225
Davies Bouldin score	0.482	0.944

Table 5.3. Topic model evaluation summary

Topic Model Algorithms	No of Topics	Coherence Score
LDA	302	0.3938
GSDMM	21	0.3782

Table 5.4. Deep learning-based clustering with TF-IDF embedding

Evaluation Methods	Clustering Algorithms	
	K-Means	HDBSCAN
Silhouette score	0.9231	0.8333
Calinski Harabasz score	43,175.8618	38,590.4055
Davies Bouldin score	0.6555	1.3484

	Text	Cluster	Hashtags	Lemma-text
0	Australia to Manufacture Covid-19 Vaccine and give it to the Citizens for free of cost: AFP quotes Prime Ministerin#CovidVaccine	0	['CovidVaccine']	australia manufacture covid vaccine and give the citizens for free cost afp quote prime minister
1	#CoronavirusVaccine #CoronaVaccine #CovidVaccine Australia is doing very good https://t.co/kBT7I6pArY	0	['CoronavirusVaccine', 'CoronaVaccine', 'CovidVaccine']	australia do very good
2	Deaths due to COVID-19 in Affected CountriesvnRead More: https://t.co/V8Y3Stu0UWvn@r_piryani @shitalbhandary... https://t.co/6jpMxX2K0l	0	NaN	deaths due covid affect countries read more
3	@Team_Subhashree @subhashreesoltwe @lamrajchoco Stay safe @subhashreesoltwe di & @lamrajchoco da ❤️❤️ ... https://t.co/ayhoaQm0Ls	0	NaN	stay safe amp
4	@michellegrattan @ConversationEDU This is what passes for leadership in our country: a voucher for something that w... https://t.co/OUUc1PeYIj	0	NaN	this what pass for leadership our country voucher for something that
5	The Multi-system Inflammatory Syndrome-Children (MIS-C) w/ #COVID19 (atypical Kawasaki disease) #COVID19India \n\nThe... https://t.co/98wj2CEkez	0	['COVID19', 'COVID19India']	the multi system inflammatory syndrome children mis atypical kawasaki disease the
6	@PrivilRodrigues @yatish57 @deepkaranahuja @shristi522 @Amrita33392520 @RashmiSrinivas14 @AkashRK_68 @SJanaQA... https://t.co/VPLa11fwfr	2	NaN	
7	@MSNBC Well, let's qualify that. would anyone of any party get a vaccine rushed out and minimally tested coming from Russia? #CovidVaccine	0	['CovidVaccine']	well let qualify that would anyone any party get vaccine rush out and minimally test come from russia
8	Most countries, without the ability to make #Vaccines locally, will be forced to rely on others like the US, China,... https://t.co/QUCozLB4Ru	0	['Vaccines']	most countries without the ability make locally will force rely others like the china
9	#DNA zooms up charts in 1st week; hear #vaccines episode: https://t.co/oDrsyhi7zN . #pandemic , #COVID19 , #CovidVaccine	0	['DNA', 'vaccines', 'pandemic', 'COVID19', 'CovidVaccine']	zoom chart week hear episode

Figure 5.31. Assigned clusters for supervised learning tasks.

6. DISCUSSION AND LIMITATIONS

6.1. Discussion

The research discussion, study limitations, and recommendations are presented in this chapter. The chapter makes an effort to explain and provide ramifications for the study's findings. The findings show that several unsupervised deep learning and machine learning techniques have been effective in identifying clusters and subjects that are common in the COVID-19 dataset. Exploratory data analysis of the coronavirus vaccine tweets provided information on the top names of individuals and organizations tweeting about the COVID-19 vaccine, user location of individuals and organizations tweeting about the COVID-19 vaccine, the devices or sources where the tweets originate from, and also keywords and time variation of the tweets. The study also demonstrates that unsupervised deep learning methods achieved better clustering results as compared to unsupervised machine learning methods. The research also supports the theory that sentence embeddings can be used to achieve better precision in these models. The evaluation metrics used in this research provided a basis for comparison between deep learning and machine learning models. The study as well demonstrated how the results of clustering could help in supervised learning tasks.

The coronavirus vaccination data included thirteen attributes, including source, user name, user description, is-retweet, user location, date, user friends, user followers, user produced, user verified, user favorites, hashtags, and text, according to exploratory data analysis. The text feature which is the main feature used in this research has 368,549 strings of text that are not null values with the data type being the object datatype. The exploratory data analysis also showed that the text field has a negligible amount of missing values, which makes the data analysis of the vaccine dataset easier to handle, as there is no need to handle missing values. The reason for missing values found in some of the attributes of the dataset is that sometimes a user does not add his/her description or location in their bio and can make a tweet without any hashtags. The result of the exploratory data analysis also showed that 99.9 percent of the text feature used in this research are unique. Further exploratory data analysis showed user name, user locations, and the source of these tweets. The results, which showed that India, the United Kingdom, the United States of America, and Canada, are the major location of users tweeting about the COVID-19 pandemic, helps to properly, define the target audience as well as the participants in this study. The results from user-name data analysis, which revealed the user name and count of tweets, suggested that many organizations are tweeting about the ongoing COVID-19 vaccination exercise. This means that governmental and non-governmental organizations are championing the course for vaccine awareness. The source of tweet results also correlated with an earlier assertion that governmental

and non-governmental organizations are championing COVID-19 awareness as about 30 percent of the tweets are from the Twitter web client, which is the application used by these organizations. The world cloud results as well supported the hypothesis that text clustering of coronavirus vaccine tweets can reveal the type of vaccine available for vaccination within a locality. As seen in the world cloud with Covishield being the predominant vaccine in India, Pfizer for Canada, Pfizer and AstraZeneca for the United Kingdom, and Pfizer for the United States of America. The results of exploratory data analysis on the time variation of the tweets defined the scope of the study. The tweets with hashtags corroborated with the topic under discourse. The exploratory data analysis results from world clouds of the tweet and other results discussed so far are in line with the hypothesis that text clustering of coronavirus vaccine tweets can reveal trends in the COVID-19 vaccination effort.

Interesting clusters and features from the coronavirus vaccine tweets dataset were discovered by machine learning algorithms from the COVID-19 vaccine tweets. k-means and HDBSCAN are two methods that were used to cluster the COVID-19 vaccination tweets. According to the results of principal component analysis (PCA) and k-means cluster labeling, there are a lot of clusters concentrated in a small area that is difficult to separate, with a few outliers that are straightforward to separate. With the exception of a few outliers that appear within and outside of clusters, data points in the t-distributed Stochastic Neighbor Embedding (t-SNE) plot were evenly distributed and easy to distinguish from one another. Clusters generated by Uniform Manifold Approximation and Projection (UMAP) seem to have labels scattered in between clusters, which makes the clusters difficult to separate. Although the data points are evenly distributed, a huge overlap of label in-between clusters makes the UMAP the worst cluster generated from these plots. Overall, the best clusters were produced by the t-SNE model utilizing the labels from the k-means clustering technique. As can be observed in the results chapter, the strongest characteristics produced by the k-means algorithm appear to be unrelated to the COVID-19 vaccination issue. A significant portion of the tweets is clustered into one group after utilizing the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) method to perform the clustering. Other minor groups were also produced. The same similar effect was evident in the results plot, where a large portion of the clusters pertaining to one group completely obscured the other clusters. The coronavirus vaccination domain appeared to be consistent with the strongest features discovered by the clustering technique, making it a better representation of the clusters. In comparison to the HDBSCAN clustering method, evaluation of the k-means clustering algorithm using the Davies Bouldin score, Calinski Harabasz score, and silhouette score often gave higher or better scores. This is to be expected as partition-based clustering algorithms optimize for distance using cluster centroids or cluster mean while Hierarchical density-based clustering algorithms optimize for areas of high density using a decision tree-like approach. Partition clustering produces

exactly k clusters, which means that whenever a point is close to the center of another cluster, it produces poor results due to data point overlapping. Hierarchical clustering, on the other hand, typically yields a much more meaningful and subjective division of clusters [60]. The best value of k in the k-means clustering algorithm was four, while the HDBCSAN algorithm produced six clusters, which are the best clusters that represent the coronavirus dataset under study.

When coronavirus vaccination tweets were clustered using deep learning, interesting clusters, and attributes that matched those of their machine learning counterparts were discovered. The COVID-19 vaccination tweets were clustered using two algorithms: HDBSCAN and k-means. The k-means cluster label findings revealed a high proportion of clusters within a region that is clearly distinguishable from one another and can be seen with the unaided eye. As can be observed in the findings chapter, the strongest features produced by the k-means clustering approach appear to be unrelated to the COVID-19 vaccination issue. A achieve, abroad, abortion, and Aaron, are some examples of features that don't have much to say regarding the COVID-19 vaccine tweets. With clusters that are quite distinct and simple to separate, k is at its best when it is six. On the contrary, clustering using the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) revealed an intriguing pattern, with a significant portion of the tweets being clustered into one group and other minor clusters also being generated, which is consistent with the outcomes from the machine learning approach. The results plot revealed the same, with the exception that there were outliers on the edges of substantial portions of the clusters that belonged to one group. The plot also showed other little clusters. However, examining the strongest features discovered by the clustering technique, the patterns seemed to be compatible with the coronavirus vaccine domain, which significantly improves its depiction of the clusters. The coronavirus vaccination tweets include elements like "vaccine", "obtain," "availability," "dosage," and "slot," among others. In comparison to the HDBSCAN clustering method, evaluation of the k-means clustering algorithm using the Davies Bouldin score, Calinski Harabasz score, and silhouette score often gave higher or better scores. This is also to be expected as partition-based clustering algorithms optimize for distance using cluster centroids or cluster mean while Hierarchical density-based clustering algorithms optimize for areas of high density using a decision tree-like approach. Because partition clustering yields exactly k clusters, if a point is near the center of another cluster, it delivers subpar results because data points overlap. Hierarchical clustering, on the other hand, typically yields a much more meaningful and subjective division of clusters [60]. The HDBCSAN algorithm produced seven clusters, which are the best clusters that represent the COVID-19 dataset under discussion. This is because HDBSCAN combines the benefits of hierarchical and density-based clustering algorithms to achieve robust clustering.

To comprehend the major themes and topics in the coronavirus vaccination dataset, topic modeling of the COVID-19 vaccine tweet was required. The topics found in the COVID-19 vaccine

corpora and the top 30 most important phrases for each topic are displayed on an inter-topic distance map created using the results of the Latent Dirichlet Allocation (LDA) based topic model. The plot is interactive and as one goes from topic one to topic two, there is a variation in the term frequency of the top 30 most salient points for each topic. The overall term frequency is the term frequency for the whole document while the estimated term frequency is the term frequency for a selected topic. The formulas for calculating the saliency term and the relevance term for the respective topic allow for easy understanding and assessment. For instance, for the whole document, the most salient term is dose followed by vaccine, get, slot, people, vaccinate, etc. The LDA-based topic model as well provided insights into the dominant topics in the document. For example, for document number zero, the dominant topic for this document is topic 202; the topic contributed 0.0172 percent to the overall document corpus. The keywords for this document are refuse, treatment, cover, cost, and so on. The text document itself is Australia manufacture the COVID-19 vaccine and gives it to citizens free of cost. This approach is the same for all documents in the corpus with varying degrees of contribution and keywords. The LDA-based topic model also revealed the top five sentences in the corpus-based on the percentage contribution of each topic to the overall document corpus. The interpretation is the same for the dominant topic data frame with the topic number, topic percentage contribution, keywords, and the associated text as features.

Results from topic modeling using the GSDMM algorithm revealed twenty-one topics for the COVID-19 vaccine corpus. The topic distribution and count of words in each topic provided a general picture of the corpus with about 403,000 words making up the corpus. The main theme for topic 1 as shown by the world cloud implied that the topic was about making an appointment and getting the first dose of the vaccine. The main theme for topic 2 as deduced from the world cloud was conversations based on beliefs, lies, and peoples' thoughts and say about the vaccine. Topic 3 consists of tweets about vaccine rollout and distribution from companies and the government. The theme of topic 4 is to raise consciousness on the COVID-19 vaccine. Words like vaccine, want, take, know, people, need, think, give, say, etc. aim to educate people and health practitioners on what they need to know before giving and taking the vaccine. Topic 5 discusses supporting and thanking the various communities, staff, and healthcare professionals that put effort into ensuring that the vaccination exercise is achieved. Topic 6 theme centered on tweets containing vaccine mandates and policies from the Government and companies issued to staff, employees, and citizens. Topic 7 salient terms talk about the risks, symptoms, and potential side effects of taking and not taking the COVID-19 vaccine. Topic 8 speaks on the need for experts to give information, answer questions, give talks, and share videos to counter disinformation on the COVID-19 vaccine. Topic 9 is made up of tweets that discuss studies to achieve herd immunity based on the strains and cases of coronavirus disease. Topic 10 solely discusses the call for people to register, schedule, and book their appointments today for the COVID-19 vaccine. Topic 11 centered on the role family, parents,

teachers, and schools have to play to ensure vaccination for children. Topic 12 dives into vaccine progress, doses administered, and the need for achieving a fully vaccinated status. Topic 13 is a small topic that has its theme on dose, slot, pay, and availability. Topic 14 encouraged people to still wear masks to stop the spread of the coronavirus disease and continue fighting the pandemic even if people are fully vaccinated to protect loved ones at home and work. Topic 15 contains tweets on phases of vaccine trials and tests by researchers, scientists, and companies to study the efficacy of vaccines and report on their safety for approval. Topic 16 discussed the general side effects of taking the covid-19 vaccine such as fatigue, arm hurt, chill, body aches, fever, pain, and so on. Topic 17 enlightens people to find a nearby clinic or vaccination site and walk to the clinic to book an appointment for a dose or booster dose of the vaccine. Topic 18 dwells on the requirements that people need to fully vaccinate to be able to travel or go to school, as well as the restrictions involved. Topic 19 talked about tweets on vaccine approval and authorization for emergency use. Topic 20 discusses the distribution and storage of vaccines for the rest of the world to prevent illness and save the rest of the world. Topic Other is a combination of buzzwords available in the COVID-19 vaccine corpus.

Topic modeling using TopicBERT / BERTopic produced some interesting patterns. Firstly, the results of the topic modeling produced seventy unique topics. Secondly, the change of topic over time showed some interesting results. Topic 0, which talks about getting the vaccine dominated discussions over time. This is to be expected as the dataset is a COVID-19 vaccine one. The most frequency of tweets on topic 0 was between December 2020 and April 2021, with the highest frequency of tweets of about 25,000. The frequency of tweets for topic 0 declined between May 2021 and September 2021and rose again from October 2021 until January 2022. The same pattern can be observed for topics 2, 3, 9, and 15 on the graph. Topics 4, 5, 13, 19, and 17 maintained a relatively flat structure over time. The hierarchical clusters generated by TopicBERT identified similar hierarchies with red and green color codes while topics that are dissimilar to one another are identified in blue color codes. Topics 0 and -1 seem to have the highest hierarchies followed by topics 11 and 48, and so on. The similarity metrics for TopicBERT showed the similarities between topics with a similarity score of 1.0 coded in the color blue meaning that the topics are highly similar while a similarity score of 0.0 coded in the color green means that the topics are dissimilar. The matrix also revealed certain degrees of similarity as seen in different hues of blue and green respectively. The topic word scores showed the percentage probability each word contributed to the topic. For instance, in topic 1, dose dose or double dose contributed the highest score, followed by dose, slot, slot dose, and free slot in that order. The same pattern is observed for the rest of the topics except topic 7 where each word contributed equal scores for the topic. The topic probability distribution for word 20 in the COVID-19 vaccine corpus showed that the word is most likely to be in topic 0 with a probability of about 0.5.

6.2. Limitations

The research on text clustering of coronavirus vaccine tweets using deep learning and machine learning algorithms as well as topic modeling of coronavirus vaccine tweets using topic-modeling algorithms produced amazing results and insights into the coronavirus vaccine tweets and as well captured the issues discussed about the coronavirus vaccine over time. Despite these results, some limitations were encountered throughout the research. These limitations may have also influenced the results of this research. In subsequent paragraphs, these limitations will be discussed in brief to understand the factors that may have affected the research.

The generalization of the results obtained in this research is limited by the data. The data obtained from this research is based on tweets with COVID-19 vaccine hashtags. This does not generalize well because some tweets made without the COVID-19 vaccine hashtags may well contain COVID-19 vaccine tweets. This also contributed to the volume of data used for this research as the comprehensive search for tweets containing COVID-19 vaccine tweets might run into millions of rows of data for the duration under study. The study also faced limited access to data as all attempts to get the raw tweets containing COVID-19 vaccine tweets without hashtags were not approved by Twitter.

The performance of clustering on COVID-19 vaccine tweets is highly dependent on the computing resources available. In this research, the resources available and time taken to perform clustering was not sufficient for a full-scale clustering of the data, hence the need to use dimensionality reduction techniques to perform clustering of the data.

There are several assumptions made by the algorithms used in this research. These algorithmic assumptions played a role in the results obtained in this research. Algorithms like k-means, HDBSCAN, UMAP, PCA, t-SNE, etc. have underlying assumptions, which directly or indirectly influenced the results of this research.

Twitter has served as a platform for citizen journalism because it was the first to report on the 2008 Mumbai bombing and the 2009 crash of an American Airlines flight. Studies done to determine the validity of tweets as an additional means of seasonal influenza surveillance revealed that Twitter performed more accurately as a seasonal influenza surveillance tool [257, 258]. However, there have been many debates by scholars on the reliability of news from the Twitter platform. Studies like detecting spammers on Twitter, misinformation on Twitter during the novel coronavirus outbreak, disinformation, etc. proved that indeed there are spammers on the Twitter platform who spread misinformation and disinformation, especially throughout the COVID-19 pandemic [259, 260]. Therefore, Twitter data suffers from information reliability issues.

7. CONCLUSIONS

This chapter will wrap up the research study by outlining the key findings in light of the study's goals and aims, as well as the study's significance and contribution to the academic community and other fields of endeavor. The study will also suggest opportunities and suggest research directions for future work.

This study used natural language processing, a variety of unsupervised machine learning and deep learning methods, algorithms, and techniques, as well as a number of topic modeling methods and algorithms to determine the ideal number of clusters and identify topics, themes, and salient terms pervasive in the coronavirus vaccine dataset. The findings show that data exploration of the tweets related to the coronavirus vaccination offered valuable information and insights into the makeup of the coronavirus vaccine dataset. Some of the insights and information obtained via exploratory data analysis include descriptive statistics of the dataset, number of unique values, number of missing values, user location, number and percentage of user names, number and percentage of devices tweeting about the coronavirus vaccine, relevant word clouds, number and count of hashtags, and count of the coronavirus vaccine tweets with respect to time. Further findings show that unsupervised deep learning clustering models achieved better performance than unsupervised machine learning clustering models on the coronavirus vaccine dataset. The study also demonstrated how sentence embeddings could be used to increase the precision of the clustering models. The top salient phrases, topics, and themes in the COVID-19 vaccination dataset were elaborated upon by topic modeling of the tweets connected to the coronavirus vaccine at various times and seasons. The topic models further show the topics discussed at a given time, topic word scores, hierarchical clusters in the topics, similarity matric, and the probability of a given word belonging to a particular topic. Although the strongest features discovered during clustering are different from this submission, evaluation of the clusters using unsupervised clustering evaluation techniques such as Davies Bouldin score, Calinski Harabasz score, and silhouette score, reveals that the K-means clustering algorithm performed better than the HDBSCAN clustering algorithm. The study also demonstrated how the results obtained by assigning clusters could be used for supervised machine learning and deep learning tasks and pattern recognition.

This research will be beneficial to health care professionals, academia as well as media houses. Health care professionals can use insights gained from the COVID-19 vaccine data as an assessment to learn lessons as well as prepare for future management and dissemination of vaccines during a pandemic. The study will also document the topics and themes that dominated the public space during the COVID-19 vaccine distribution. This study, which is a novel study, will close the existing gap in COVID-19 vaccine research in academia especially as it relates to unsupervised

learning. The data used in this research will also be available as a resource for future academic research and analysis and the results from this research will act as a benchmark for pandemic vaccine tweets in the future. Media houses can use the insights gained to support decision-making in the future management of a pandemic to prevent the proliferation of fake news and ensure that the correct information is disseminated to the correct audience. The models generated in this study can be used to automatically obtain document clusters and extract topics from a document corpus for various applications.

There are certain limitations experienced during this research. The data used for this research suffer from sampling issues, which can affect the generalizability of the research. This is because the data was collected strictly from tweets containing the COVID-19 vaccine hashtags. Access to high computing resources that can improve the accuracy of the models was not available. The lack of a previous study on this research area caused various constraints on time, and money and might have introduced researcher bias. However, the research was conducted in a professional manner using available tools, techniques, and algorithms available in the literature. This is can be seen in the accuracy of the clusters and the topics generated from the research.

8. RECOMMENDATIONS

Further research using keyword searches of the COVID-19 vaccine is recommended to obtain better generalizability of the results. This can be achieved by gaining access to the Twitter API for academia. More research is also encouraged using other clustering algorithms other than K-means or HDBSCAN to gauge performance. Research on detecting outliers in the COVID-19 vaccine dataset is an interesting area of research that can be explored to provide better insights. Generally, more research is needed in the area of unsupervised learning in the COVID-19 domain to assist in creating more knowledge on COVID-19. For instance, research on how to create word embeddings particularly suitable for COVID-19 and general health tweets is highly recommended to provide context-aware embedding for supervised and unsupervised learning tasks. The insights and information gotten from Twitter should be used with caution. It is recommended that twitter data be used in conjunction with other quality datasets to improve any application because of information integrity issues with Twitter data.

REFERENCES

[1] M. Rosell, "Text Clustering Exploration – Swedish Text Representation and Clustering Results Unraveled," 2009.

[2] "Coronavirus." https://www.who.int/westernpacific/health-topics/coronavirus (accessed Feb. 07, 2022).

[3] "Coronavirus (COVID-19) Overview," *WebMD*. https://www.webmd.com/lung/coronavirus (accessed Feb. 07, 2022).

[4] H. Ritchie *et al.*, "Coronavirus Pandemic (COVID-19)," *Our World Data*, Mar. 2020, Accessed: Feb. 07, 2022. [Online]. Available: https://ourworldindata.org/covid-vaccinations

[5] D. O. Ukwen and M. Karabatak, "Review of NLP-based Systems in Digital Forensics and Cybersecurity," in *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, Jun. 2021, pp. 1–9. doi: 10.1109/ISDFS52919.2021.9486354.

[6] "How much data is generated each day?," *World Economic Forum*. https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/ (accessed Feb. 07, 2022).

[7] "Introduction to Machine Learning," p. 456.

[8] G. C. Nutakki, B. Abdollahi, W. Sun, and O. Nasraoui, "An Introduction to Deep Clustering," in *Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications*, O. Nasraoui and C.-E. Ben N'Cir, Eds. Cham: Springer International Publishing, 2019, pp. 73–89. doi: 10.1007/978-3-319-97864-2_4.

[9] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," *ACM SIGIR Forum*, vol. 51, no. 2, pp. 148–159, Aug. 2017, doi: 10.1145/3130348.3130362.

[10] C. J. van Rijsbergen, "INFORMATION RETRIEVAL : THEORY AND PRACTICE," p. 14.

[11] B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2017, pp. 745–750. doi: 10.1109/ICCONS.2017.8250563.

[12] R. Vaishya, M. Javaid, I. Haleem Khan, and A. Haleem, "Artificial Intelligence (AI) applications for COVID-19 pandemic," *Diabetes Metab. Syndr. Clin. Res. Rev.*, vol. 14, Apr. 2020, doi: 10.1016/j.dsx.2020.04.012.

[13] D. Folkz, "How Artificial Intelligence, Data Science And prominent technologies are used to fight against the…," *Medium*, Aug. 17, 2020. https://medium.com/@Datafolkz_blogs/how-artificial-intelligence-data-science-and-prominent-technologies-are-used-to-fight-against-the-29f77d5bd906 (accessed Feb. 08, 2022).

[14] "What is Twitter, a social network or a news media? | Proceedings of the 19th international conference on World wide web." https://dl.acm.org/doi/abs/10.1145/1772690.1772751 (accessed Feb. 08, 2022).

[15] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, New York, NY, USA, Aug. 2007, pp. 56–65. doi: 10.1145/1348549.1348556.

[16] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, "Who says what to whom on twitter," in *Proceedings of the 20th international conference on World wide web*, New York, NY, USA, Mar. 2011, pp. 705–714. doi: 10.1145/1963405.1963504.

[17] M. Paul and M. Dredze, "You Are What You Tweet: Analyzing Twitter for Public Health," *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 5, no. 1, Art. no. 1, 2011.

[18] S. Kaur, P. Kaul, and P. Moradian Zadeh, "Monitoring the Dynamics of Emotions during COVID-19 Using Twitter Data," *Procedia Comput. Sci.*, vol. 177, pp. 423–430, Jan. 2020, doi: 10.1016/j.procs.2020.10.056.

[19] M. O. Lwin *et al.*, "Global Sentiments Surrounding the COVID-19 Pandemic on Twitter: Analysis of Twitter Trends," *JMIR Public Health Surveill.*, vol. 6, no. 2, p. e19447, May 2020, doi: 10.2196/19447.

[20] A. Abd-Alrazaq, D. Alhuwail, M. Househ, M. Hamdi, and Z. Shah, "Top Concerns of Tweeters During the COVID-19 Pandemic: Infoveillance Study," *J. Med. Internet Res.*, vol. 22, no. 4, p. e19016, Apr. 2020, doi: 10.2196/19016.

[21] Y. Pershad, P. T. Hangge, H. Albadawi, and R. Oklu, "Social Medicine: Twitter in Healthcare," *J. Clin. Med.*, vol. 7, no. 6, Art. no. 6, Jun. 2018, doi: 10.3390/jcm7060121.

[22] S. E. Jordan, S. E. Hovet, I. C.-H. Fung, H. Liang, K.-W. Fu, and Z. T. H. Tse, "Using Twitter for Public Health Surveillance from Monitoring and Prediction to Public Response," *Data*, vol. 4, no. 1, Art. no. 1, Mar. 2019, doi: 10.3390/data4010006.
[23] "Journal of Medical Internet Research - COVID-19 Vaccine–Related Discussion on Twitter: Topic Modeling and Sentiment Analysis." https://www.jmir.org/2021/6/e24435/ (accessed Feb. 08, 2022).
[24] H.-H. Bock, "Clustering Methods: A History of k-Means Algorithms," in *Selected Contributions in Data Analysis and Classification*, P. Brito, G. Cucumel, P. Bertrand, and F. de Carvalho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 161–172. doi: 10.1007/978-3-540-73560-1_15.
[25] L. V. Bijuraj, "Clustering and its Applications," p. 4, 2013.
[26] R. C. T. Lee, "Clustering Analysis and Its Applications," in *Advances in Information Systems Science: Volume 8*, J. T. Tou, Ed. Boston, MA: Springer US, 1981, pp. 169–292. doi: 10.1007/978-1-4613-9883-7_4.
[27] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999, doi: 10.1145/331499.331504.
[28] D. Wunsch and R. Xu, *Clustering*. John Wiley & Sons, 2008.
[29] U. von Luxburg, R. C. Williamson, and I. Guyon, "Clustering: Science or Art?," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Jun. 2012, pp. 65–79. Accessed: Dec. 20, 2021. [Online]. Available: https://proceedings.mlr.press/v27/luxburg12a.html
[30] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.
[31] C. Ma and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, vol. 20. 2007. doi: 10.1137/1.9780898718348.
[32] G. W. Milligan and M. C. Cooper, "Methodology Review: Clustering Methods," *Appl. Psychol. Meas.*, vol. 11, no. 4, pp. 329–354, Dec. 1987, doi: 10.1177/014662168701100401.
[33] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intell. Data Anal.*, vol. 11, no. 6, pp. 583–605, Jan. 2007, doi: 10.3233/IDA-2007-11602.
[34] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967, doi: 10.1007/BF02289588.
[35] F. Murtagh, "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, Nov. 1983, doi: 10.1093/comjnl/26.4.354.
[36] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 86–97, 2012, doi: 10.1002/widm.53.
[37] F. Murtagh and P. Contreras, "Methods of Hierarchical Clustering," *ArXiv11050121 Cs Math Stat*, Apr. 2011, Accessed: Dec. 20, 2021. [Online]. Available: http://arxiv.org/abs/1105.0121
[38] P. Berkhin, "A Survey of Clustering Data Mining Techniques," in *Grouping Multidimensional Data: Recent Advances in Clustering*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Berlin, Heidelberg: Springer, 2006, pp. 25–71. doi: 10.1007/3-540-28349-8_2.
[39] H. Pirim, *Recent Applications in Data Clustering*. BoD – Books on Demand, 2018.
[40] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Min. Knowl. Discov.*, vol. 1, no. 3, pp. 231–240, 2011, doi: 10.1002/widm.30.
[41] M. Ester, "Density-Based Clustering," in *Data Clustering*, Chapman and Hall/CRC, 2014.
[42] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Min. Knowl. Discov.*, vol. 10, no. 2, p. e1343, 2020, doi: 10.1002/widm.1343.
[43] "Mercer kernel-based clustering in feature space | IEEE Journals & Magazine | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/1000150/ (accessed Dec. 22, 2021).
[44] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 801–805, May 2005, doi: 10.1109/TPAMI.2005.88.
[45] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 176–190, Jan. 2008, doi: 10.1016/j.patcog.2007.05.018.
[46] C. Piciarelli, C. Micheloni, and G. l. Foresti, "Kernel-based clustering," *Electron. Lett.*, vol. 49, no. 2, pp. 113–114, 2013, doi: 10.1049/el.2012.3234.
[47] H.-H. Bock, "Clustering and Neural Network Approaches," in *Classification in the Information Age*, Berlin, Heidelberg, 1999, pp. 42–57. doi: 10.1007/978-3-642-60187-3_4.
[48] K.-L. Du, "Clustering: A neural network approach," *Neural Netw.*, vol. 23, no. 1, pp. 89–107, Jan. 2010, doi: 10.1016/j.neunet.2009.08.007.
[49] F. Roohi, "Artificial Neural Network Approach to Clustering," p. 6.
[50] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2002, vol. 14. Accessed: Dec. 22, 2021. [Online].

Available: https://proceedings.neurips.cc/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html
[51] F. R. Bach and M. I. Jordan, "Learning Spectral Clustering," p. 14.
[52] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007, doi: 10.1007/s11222-007-9033-z.
[53] J. Liu and J. Han, "Spectral Clustering," in *Data Clustering*, Chapman and Hall/CRC, 2014.
[54] "PII: S0019-9958(69)90591-9 | Elsevier Enhanced Reader." https://reader.elsevier.com/reader/sd/pii/S0019995869905919?token=7970002C260CFF311B3918CDF77A78BD1390E437AF0D257CA0C4C90704560B41161426EBB3D4CB2588DB6CD2E0ADE7CB&originRegion=eu-west-1&originCreation=20211223083434 (accessed Dec. 23, 2021).
[55] E. H. Ruspini, "Numerical methods for fuzzy clustering," *Inf. Sci.*, vol. 2, no. 3, pp. 319–350, Jul. 1970, doi: 10.1016/S0020-0255(70)80056-1.
[56] M.-S. Yang, "A survey of fuzzy clustering," *Math. Comput. Model.*, vol. 18, no. 11, pp. 1–16, Dec. 1993, doi: 10.1016/0895-7177(93)90202-A.
[57] J. V. de Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*. John Wiley & Sons, 2007.
[58] G. Fung, "A Comprehensive Overview of Basic Clustering Algorithms," p. 37.
[59] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.
[60] O. Abu Abbas, "Comparisons Between Data Clustering Algorithms," *Int Arab J Inf Technol*, vol. 5, pp. 320–325, Jul. 2008.
[61] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015, doi: 10.1007/s40745-015-0040-1.
[62] M. Z. Rodriguez *et al.*, "Clustering algorithms: A comparative approach," *PLOS ONE*, vol. 14, no. 1, p. e0210236, Jan. 2019, doi: 10.1371/journal.pone.0210236.
[63] A. Fahad *et al.*, "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 3, pp. 267–279, Sep. 2014, doi: 10.1109/TETC.2014.2330519.
[64] R. S. M. L. Patibandla and N. Veeranjaneyulu, "Survey on Clustering Algorithms for Unstructured Data," in *Intelligent Engineering Informatics*, Singapore, 2018, pp. 421–429. doi: 10.1007/978-981-10-7566-7_41.
[65] S. A. Elavarasi, "A SURVEY ON PARTITION CLUSTERING ALGORITHMS," vol. 1, no. 1, p. 14, 2011.
[66] V. Thambusamy and S. T, "A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach," *Inf. Technol. J.*, vol. 10, Mar. 2011, doi: 10.3923/itj.2011.478.484.
[67] S. Na, L. Xumin, and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, Apr. 2010, pp. 63–67. doi: 10.1109/IITSI.2010.74.
[68] J. Wang and X. Su, "An improved K-Means clustering algorithm," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, May 2011, pp. 44–46. doi: 10.1109/ICCSN.2011.6014384.
[69] I. Mohamad and D. Usman, "Standardization and Its Effects on K-Means Clustering Algorithm," *Res. J. Appl. Sci. Eng. Technol.*, vol. 6, pp. 3299–3303, Sep. 2013, doi: 10.19026/rjaset.6.3638.
[70] T. Kodinariya and P. Makwana, "Review on Determining of Cluster in K-means Clustering," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 1, pp. 90–95, Jan. 2013.
[71] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2009.
[72] R. T. Ng and J. Han, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002, doi: 10.1109/TKDE.2002.1033770.
[73] D. P. Dabhi and M. R. Patel, "Extensive Survey on Hierarchical Clustering Methods in Data Mining," vol. 03, no. 11, p. 8.
[74] P. Contreras and F. Murtagh, "Hierarchical Clustering," in *Handbook of Cluster Analysis*, Chapman and Hall/CRC, 2015.
[75] M.-F. Balcan, Y. Liang, and P. Gupta, "Robust Hierarchical Clustering," p. 41.
[76] C. Ding and X. He, "Cluster merging and splitting in hierarchical clustering algorithms," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, Dec. 2002, pp. 139–146. doi: 10.1109/ICDM.2002.1183896.
[77] Y. Rani and H. Rohil, "A Study of Hierarchical Clustering Algorithm," p. 8.

[78] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky, "Automatic Extraction of Clusters from Hierarchical Clustering Representations," in *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, 2003, pp. 75–87. doi: 10.1007/3-540-36175-8_8.

[79] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *16th IEEE International Conference on Tools with Artificial Intelligence*, Nov. 2004, pp. 576–584. doi: 10.1109/ICTAI.2004.50.

[80] S. Dasgupta, "A cost function for similarity-based hierarchical clustering," in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, New York, NY, USA, Jun. 2016, pp. 118–127. doi: 10.1145/2897518.2897527.

[81] V. Cohen-addad, V. Kanade, F. Mallmann-trenn, and C. Mathieu, "Hierarchical Clustering: Objective Functions and Algorithms," *J. ACM*, vol. 66, no. 4, p. 26:1-26:42, Jun. 2019, doi: 10.1145/3321386.

[82] W.-K. Loh and Y.-H. Park, "A Survey on Density-Based Clustering Algorithms," in *Ubiquitous Information Technologies and Applications*, Berlin, Heidelberg, 2014, pp. 775–780. doi: 10.1007/978-3-642-41671-2_98.

[83] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Front. Comput. Sci.*, vol. 15, no. 1, p. 151308, Sep. 2020, doi: 10.1007/s11704-019-9059-3.

[84] P. B. Nagpal, N. Kurukshetra, and P. A. Mann, "Comparative Study of Density based Clustering Algorithms."

[85] A. Amini, T. Y. Wah, and H. Saboohi, "On Density-Based Data Streams Clustering Algorithms: A Survey," *J. Comput. Sci. Technol.*, vol. 29, no. 1, pp. 116–141, Jan. 2014, doi: 10.1007/s11390-014-1416-y.

[86] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Aug. 2007, pp. 133–142. doi: 10.1145/1281192.1281210.

[87] "A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases."

[88] M. Ester, H.-P. Kriegel, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," p. 6.

[89] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, Jun. 1998, doi: 10.1023/A:1009745219419.

[90] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, 2013, pp. 160–172. doi: 10.1007/978-3-642-37456-2_14.

[91] H.-P. Kriegel and M. Pfeifle, "Hierarchical density-based clustering of uncertain data," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Nov. 2005, p. 4 pp.-. doi: 10.1109/ICDM.2005.75.

[92] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. I," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 29, no. 6, pp. 778–785, Dec. 1999, doi: 10.1109/3477.809032.

[93] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. II," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 29, no. 6, pp. 786–801, Dec. 1999, doi: 10.1109/3477.809033.

[94] M. A. Al-Nimr, "THE INTERNATIONAL ADVISORY BOARD," p. 108.

[95] A. Gosain and S. Dahiya, "Performance Analysis of Various Fuzzy Clustering Algorithms: A Review," *Procedia Comput. Sci.*, vol. 79, pp. 100–111, Jan. 2016, doi: 10.1016/j.procs.2016.03.014.

[96] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2, pp. 191–203, Jan. 1984, doi: 10.1016/0098-3004(84)90020-7.

[97] S. Miyamoto, H. Ichihashi, and K. Honda, "Introduction," in *Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications*, S. Miyamoto, H. Ichihashi, and K. Honda, Eds. Berlin, Heidelberg: Springer, 2008, pp. 1–7. doi: 10.1007/978-3-540-78737-2_1.

[98] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 773–780, Jul. 1989, doi: 10.1109/34.192473.

[99] M. H. Ghaseminezhad and A. Karami, "A novel self-organizing map (SOM) neural network for discrete groups of data clustering," *Appl. Soft Comput.*, vol. 11, no. 4, pp. 3771–3778, Jun. 2011, doi: 10.1016/j.asoc.2011.02.009.

[100] H. Liu and X. Ban, "Clustering by growing incremental self-organizing neural network," *Expert Syst. Appl.*, vol. 42, no. 11, pp. 4965–4981, Jul. 2015, doi: 10.1016/j.eswa.2015.02.006.

[101] C.-C. Chang and S.-H. Chen, “A comparative analysis on artificial neural network-based two-stage clustering,” *Cogent Eng.*, vol. 2, no. 1, p. 995785, Dec. 2015, doi: 10.1080/23311916.2014.995785.
[102] M. Han and J. Xi, “Efficient clustering of radial basis perceptron neural network for pattern recognition,” *Pattern Recognit.*, vol. 37, no. 10, pp. 2059–2067, Oct. 2004, doi: 10.1016/j.patcog.2004.02.014.
[103] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, “Approximate kernel k-means: solution to large scale kernel clustering,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Aug. 2011, pp. 895–903. doi: 10.1145/2020408.2020558.
[104] B. Zhao, J. T. Kwok, and C. Zhang, “Multiple Kernel Clustering,” in *Proceedings of the 2009 SIAM International Conference on Data Mining (SDM)*, Society for Industrial and Applied Mathematics, 2009, pp. 638–649. doi: 10.1137/1.9781611972795.55.
[105] A. K. Qin and P. N. Suganthan, “Kernel neural gas algorithms with application to cluster analysis,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Aug. 2004, vol. 4, pp. 617-620 Vol.4. doi: 10.1109/ICPR.2004.1333848.
[106] D.-Q. Zhang and S.-C. Chen, “Clustering Incomplete Data Using Kernel-Based Fuzzy C-means Algorithm,” *Neural Process. Lett.*, vol. 18, no. 3, pp. 155–162, Dec. 2003, doi: 10.1023/B:NEPL.0000011135.19145.1b.
[107] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, “Multiple Kernel Fuzzy Clustering,” *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, Feb. 2012, doi: 10.1109/TFUZZ.2011.2170175.
[108] S. Ding, L. Zhang, and Y. Zhang, “Research on Spectral Clustering algorithms and prospects,” in *2010 2nd International Conference on Computer Engineering and Technology*, Apr. 2010, vol. 6, pp. V6-149-V6-153. doi: 10.1109/ICCET.2010.5486345.
[109] D. Verma and M. Meil, “A Comparison of Spectral Clustering Algorithms,” p. 19.
[110] D. Yan, L. Huang, and M. I. Jordan, “Fast approximate spectral clustering,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Jun. 2009, pp. 907–916. doi: 10.1145/1557019.1557118.
[111] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel Spectral Clustering in Distributed Systems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 568–586, Mar. 2011, doi: 10.1109/TPAMI.2010.88.
[112] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel Spectral Clustering,” in *Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg, 2008, pp. 374–389. doi: 10.1007/978-3-540-87481-2_25.
[113] “On evolutionary spectral clustering | ACM Transactions on Knowledge Discovery from Data.” https://dl.acm.org/doi/abs/10.1145/1631162.1631165 (accessed Jan. 06, 2022).
[114] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, “On evolutionary spectral clustering,” *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 4, p. 17:1-17:30, Dec. 2009, doi: 10.1145/1631162.1631165.
[115] M. B. Blaschko and C. H. Lampert, “Correlational spectral clustering,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587353.
[116] E. Elhamifar and R. Vidal, “Sparse Subspace Clustering: Algorithm, Theory, and Applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013, doi: 10.1109/TPAMI.2013.57.
[117] H. D. Iii, “A Co-training Approach for Multi-view Spectral Clustering Abhishek Kumar.”
[118] A. Kumar, P. Rai, and H. D. Iii, “Co-regularized Multi-view Spectral Clustering,” p. 9.
[119] R. H. Sheikh, M. M. Raghuwanshi, and A. N. Jaiswal, “Genetic Algorithm Based Clustering: A Survey,” in *2008 First International Conference on Emerging Trends in Engineering and Technology*, Jul. 2008, pp. 314–319. doi: 10.1109/ICETET.2008.48.
[120] U. Maulik and S. Bandyopadhyay, “Genetic algorithm-based clustering technique,” *Pattern Recognit.*, vol. 33, no. 9, pp. 1455–1465, Sep. 2000, doi: 10.1016/S0031-3203(99)00137-5.
[121] G. Garai and B. B. Chaudhuri, “A novel genetic algorithm for automatic clustering,” *Pattern Recognit. Lett.*, vol. 25, no. 2, pp. 173–187, Jan. 2004, doi: 10.1016/j.patrec.2003.09.012.
[122] L. E. Agustı´n-Blas, S. Salcedo-Sanz, S. Jiménez-Fernández, L. Carro-Calvo, J. Del Ser, and J. A. Portilla-Figueras, “A new grouping genetic algorithm for clustering problems,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9695–9703, Aug. 2012, doi: 10.1016/j.eswa.2012.02.149.
[123] A. A. A. Esmin, R. A. Coelho, and S. Matwin, “A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data,” *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 23–45, Jun. 2015, doi: 10.1007/s10462-013-9400-4.

[124] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. Ur Rehman, “Research on particle swarm optimization based clustering: A systematic review of literature and techniques,” *Swarm Evol. Comput.*, vol. 17, pp. 1–13, Aug. 2014, doi: 10.1016/j.swevo.2014.02.001.
[125] S. Rana, S. Jasola, and R. Kumar, “A review on particle swarm optimization algorithms and their applications to data clustering,” *Artif. Intell. Rev.*, vol. 35, no. 3, pp. 211–222, Mar. 2011, doi: 10.1007/s10462-010-9191-9.
[126] D. W. van der Merwe and A. P. Engelbrecht, “Data clustering using particle swarm optimization,” in *The 2003 Congress on Evolutionary Computation, 2003. CEC ’03.*, Dec. 2003, vol. 1, pp. 215-220 Vol.1. doi: 10.1109/CEC.2003.1299577.
[127] I. Aljarah and S. A. Ludwig, “Parallel particle swarm optimization clustering algorithm based on MapReduce methodology,” in *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*, Nov. 2012, pp. 104–111. doi: 10.1109/NaBIC.2012.6402247.
[128] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, “Integration of particle swarm optimization and genetic algorithm for dynamic clustering,” *Inf. Sci.*, vol. 195, pp. 124–140, Jul. 2012, doi: 10.1016/j.ins.2012.01.021.
[129] R. J. Kuo and L. M. Lin, “Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering,” *Decis. Support Syst.*, vol. 49, no. 4, pp. 451–462, Nov. 2010, doi: 10.1016/j.dss.2010.05.006.
[130] D. T. Pham and A. A. Afify, “Clustering techniques and their applications in engineering,” *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 221, no. 11, pp. 1445–1459, Nov. 2007, doi: 10.1243/09544062JMES508.
[131] H. Frigui and R. Krishnapuram, “A robust competitive clustering algorithm with applications in computer vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999, doi: 10.1109/34.765656.
[132] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*. Springer Science & Business Media, 2011.
[133] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications,” in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, New York, NY, USA, Jun. 1998, pp. 94–105. doi: 10.1145/276304.276314.
[134] A. Dharmarajan and T. Velmurugan, “Applications of partition based clustering algorithms: A survey,” in *2013 IEEE International Conference on Computational Intelligence and Computing Research*, Dec. 2013, pp. 1–5. doi: 10.1109/ICCIC.2013.6724235.
[135] J. Li and H. W. Lewis, “Fuzzy Clustering Algorithms — Review of the Applications,” in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, Nov. 2016, pp. 282–288. doi: 10.1109/SmartCloud.2016.14.
[136] Z. Li *et al.*, “Convolutional Neural Network Based Clustering and Manifold Learning Method for Diabetic Plantar Pressure Imaging Dataset,” *J. Med. Imaging Health Inform.*, vol. 7, no. 3, pp. 639–652, Jun. 2017, doi: 10.1166/jmihi.2017.2082.
[137] A. A. Frolov, D. Husek, and P. Yu. Polyakov, “Recurrent-Neural-Network-Based Boolean Factor Analysis and Its Application to Word Clustering,” *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1073–1086, Jul. 2009, doi: 10.1109/TNN.2009.2016090.
[138] S. K. Rangarajan, V. V. Phoha, K. S. Balagani, R. R. Selmic, and S. S. Iyengar, “Adaptive neural network clustering of Web users,” *Computer*, vol. 37, no. 4, pp. 34–40, Apr. 2004, doi: 10.1109/MC.2004.1297299.
[139] N. K. Nehra, M. Kumar, and R. B. Patel, “Neural Network Based Energy Efficient Clustering and Routing in Wireless Sensor Networks,” in *2009 First International Conference on Networks Communications*, Dec. 2009, pp. 34–39. doi: 10.1109/NetCoM.2009.56.
[140] L. Duan, L. Xu, F. Guo, J. Lee, and B. Yan, “A local-density based spatial clustering algorithm with noise,” *Inf. Syst.*, vol. 32, no. 7, pp. 978–986, Nov. 2007, doi: 10.1016/j.is.2006.10.006.
[141] Y. El-Sonbaty, M. A. Ismail, and M. Farouk, “An efficient density based clustering algorithm for large databases,” in *16th IEEE International Conference on Tools with Artificial Intelligence*, Nov. 2004, pp. 673–677. doi: 10.1109/ICTAI.2004.27.
[142] M. E. Celebi, Y. A. Aslandogan, and P. R. Bergstresser, “Mining biomedical images with density-based clustering,” in *International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II*, Apr. 2005, vol. 1, pp. 163-168 Vol. 1. doi: 10.1109/ITCC.2005.196.

[143] D. Jiang, J. Pei, and A. Zhang, "DHC: a density-based hierarchical clustering method for time series gene expression data," in *Third IEEE Symposium on Bioinformatics and Bioengineering, 2003. Proceedings.*, Mar. 2003, pp. 393–400. doi: 10.1109/BIBE.2003.1188978.
[144] C.-H. Wang, "Outlier identification and market segmentation using kernel-based clustering techniques," *Expert Syst. Appl.*, vol. 36, no. 2, Part 2, pp. 3744–3750, Mar. 2009, doi: 10.1016/j.eswa.2008.02.037.
[145] L. Liao, D. Wang, F. Wang, and L. Yuan, "A fast kernel-based clustering algorithm with application in MRI image segmentation," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Nov. 2009, vol. 4, pp. 405–410. doi: 10.1109/ICICISYS.2009.5357645.
[146] H. Amadou Boubacar, S. Lecoeuche, and S. Maouche, "SAKM: Self-adaptive kernel machine A kernel-based algorithm for online clustering," *Neural Netw.*, vol. 21, no. 9, pp. 1287–1301, Nov. 2008, doi: 10.1016/j.neunet.2008.03.016.
[147] L. Liao and T.-S. Lin, "A fast spatial constrained fuzzy kernel clustering algorithm for MRI brain image segmentation," in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, Nov. 2007, vol. 1, pp. 82–87. doi: 10.1109/ICWAPR.2007.4420641.
[148] G. Li, S. Xu, J. Wu, and H. Ding, "Resource Scheduling Based on Improved Spectral Clustering Algorithm in Edge Computing," *Sci. Program.*, vol. 2018, p. e6860359, Jul. 2018, doi: 10.1155/2018/6860359.
[149] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Min. Knowl. Discov.*, vol. 28, no. 1, pp. 1–30, Jan. 2014, doi: 10.1007/s10618-012-0291-9.
[150] T. Schultz and G. L. Kindlmann, "Open-Box Spectral Clustering: Applications to Medical Image Analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2100–2108, Dec. 2013, doi: 10.1109/TVCG.2013.181.
[151] D. J. Higham, G. Kalna, and M. Kibble, "Spectral clustering and its use in bioinformatics," *J. Comput. Appl. Math.*, vol. 204, no. 1, pp. 25–37, Jul. 2007, doi: 10.1016/j.cam.2006.04.026.
[152] Z. Li and J. Chen, "Superpixel Segmentation Using Linear Spectral Clustering," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1356–1363. Accessed: Jan. 10, 2022. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2015/html/Li_Superpixel_Segmentation_Using_2015_CVPR_paper.html
[153] R. Liu and H. Zhang, "Segmentation of 3D meshes through spectral clustering," in *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, Oct. 2004, pp. 298–305. doi: 10.1109/PCCGA.2004.1348360.
[154] A. Ekin, S. Pankanti, and A. Hampapur, "Initialization-independent spectral clustering with applications to automatic video analysis," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004, vol. 3, pp. iii–641. doi: 10.1109/ICASSP.2004.1326626.
[155] Z. Wang, B. Liu, S. Chen, S. Ma, L. Xue, and H. Zhao, "A Manifold Proximal Linear Method for Sparse Spectral Clustering with Application to Single-Cell RNA Sequencing Data Analysis," *Inf. J. Optim.*, Nov. 2021, doi: 10.1287/ijoo.2021.0064.
[156] W. Pentney, "Spectral Clustering of Biological Sequence Data," p. 6.
[157] A. R. Benson, D. F. Gleich, and J. Leskovec, "Tensor Spectral Clustering for Partitioning Higher-order Network Structures," in *Proceedings of the 2015 SIAM International Conference on Data Mining (SDM)*, Society for Industrial and Applied Mathematics, 2015, pp. 118–126. doi: 10.1137/1.9781611974010.14.
[158] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Math. Probl. Eng.*, vol. 2015, p. e931256, Oct. 2015, doi: 10.1155/2015/931256.
[159] V. Gupta and G. S. Lehal, "A Survey of Text Mining Techniques and Applications," *J. Emerg. Technol. Web Intell.*, vol. 1, no. 1, pp. 60–76, Aug. 2009, doi: 10.4304/jetwi.1.1.60-76.
[160] A. Hotho, A. Nurnberger, G. Paaß, and S. Augustin, "A Brief Survey of Text Mining," p. 37.
[161] V. Preamsudha, "A Survey on Various Approaches in Document Clustering K.Sathiyakumari,."
[162] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, "An Evaluation on Feature Selection for Text Clustering," p. 8.
[163] L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on unsupervised feature selection methods for text clustering," in *2005 International Conference on Natural Language Processing and Knowledge Engineering*, Oct. 2005, pp. 597–601. doi: 10.1109/NLPKE.2005.1598807.
[164] L. M. Abualigah and A. T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," *J. Supercomput.*, vol. 73, no. 11, pp. 4773–4795, Nov. 2017, doi: 10.1007/s11227-017-2046-2.

[165] L. M. Abualigah, A. T. Khader, M. A. Al-Betar, and O. A. Alomari, “Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering,” *Expert Syst. Appl.*, vol. 84, pp. 24–36, Oct. 2017, doi: 10.1016/j.eswa.2017.05.002.
[166] H. Liang, X. Sun, Y. Sun, and Y. Gao, “Text feature extraction based on deep learning: a review,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2017, no. 1, p. 211, Dec. 2017, doi: 10.1186/s13638-017-0993-1.
[167] A. I. Kadhim, Y.-N. Cheah, and N. H. Ahamed, “Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering,” in *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, Dec. 2014, pp. 69–73. doi: 10.1109/ICAIET.2014.21.
[168] V. Angelis, G. Felici, and G. Mancinelli, “Feature Selection for Data Mining,” in *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, vol. 6, 2006, pp. 227–252. doi: 10.1007/0-387-34296-6_6.
[169] A. K. Nikhath and K. Subrahmanyam, “Feature selection, optimization and clustering strategies of text documents,” *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 2, p. 1313, Apr. 2019, doi: 10.11591/ijece.v9i2.pp1313-1320.
[170] N. Shah and S. Mahajan, “Semantic based Document Clustering: A Detailed Review.”
[171] A. Hotho, A. Maedche, and S. Staab, “Ontology-based Text Document Clustering.,” *KI*, vol. 16, pp. 48–54, Jan. 2002.
[172] A. Hotho, S. Staab, and G. Stumme, “Ontologies improve text document clustering,” in *Third IEEE International Conference on Data Mining*, Nov. 2003, pp. 541–544. doi: 10.1109/ICDM.2003.1250972.
[173] C. Biemann, “Ontology Learning from Text: A Survey of Methods,” *LDV-Forum*, vol. 20, pp. 75–93, Jan. 2005.
[174] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer Science & Business Media, 2006.
[175] S. C. Punitha and M. Punithavalli, “Performance Evaluation of Semantic Based and Ontology Based Text Document Clustering Techniques,” *Procedia Eng.*, vol. 30, pp. 100–106, Jan. 2012, doi: 10.1016/j.proeng.2012.01.839.
[176] F. Beil, M. Ester, and X. Xu, “Frequent term-based text clustering,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Jul. 2002, pp. 436–442. doi: 10.1145/775047.775110.
[177] Y. Li, S. M. Chung, and J. D. Holt, “Text document clustering based on frequent word meaning sequences,” *Data Knowl. Eng.*, vol. 64, no. 1, pp. 381–404, Jan. 2008, doi: 10.1016/j.datak.2007.08.001.
[178] C. Luo, Y. Li, and S. M. Chung, “Text document clustering based on neighbors,” *Data Knowl. Eng.*, vol. 68, no. 11, pp. 1271–1288, Nov. 2009, doi: 10.1016/j.datak.2009.06.007.
[179] V. Mehta, S. Bawa, and J. Singh, “WEClustering: word embeddings based text clustering technique for large datasets,” *Complex Intell. Syst.*, vol. 7, no. 6, pp. 3211–3224, Dec. 2021, doi: 10.1007/s40747-021-00512-9.
[180] S. Shehata, F. Karray, and M. Kamel, “An Efficient Concept-Based Mining Model for Enhancing Text Clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1360–1371, Oct. 2010, doi: 10.1109/TKDE.2009.174.
[181] A. Hotho and G. Stumme, “Conceptual Clustering of Text Clusters,” p. 9.
[182] J. R and W. J. G, “A Survey on Optimization Approaches to Text Document Clustering,” *Int. J. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 31–44, Dec. 2013, doi: 10.5121/ijcsa.2013.3604.
[183] F. Liu and L. Xiong, “Survey on text clustering algorithm -Research present situation of text clustering algorithm,” in *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, Jul. 2011, pp. 196–199. doi: 10.1109/ICSESS.2011.5982288.
[184] C. C. Aggarwal and C. Zhai, “A Survey of Text Clustering Algorithms,” in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 77–128. doi: 10.1007/978-1-4614-3223-4_4.
[185] B. Larsen and C. Aone, “Fast and effective text mining using linear-time document clustering,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Aug. 1999, pp. 16–22. doi: 10.1145/312129.312186.
[186] P. Pantel and D. Lin, “Discovering word senses from text,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Jul. 2002, pp. 613–619. doi: 10.1145/775047.775138.

[187] K. Kummamuru, A. Dhawale, and R. Krishnapuram, “Fuzzy co-clustering of documents and keywords,” in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ ’03.*, May 2003, vol. 2, pp. 772–777 vol.2. doi: 10.1109/FUZZ.2003.1206527.
[188] K. M. Hammouda, D. N. Matute, and M. S. Kamel, “CorePhrase: Keyphrase Extraction for Document Clustering,” in *Machine Learning and Data Mining in Pattern Recognition*, Berlin, Heidelberg, 2005, pp. 265–274. doi: 10.1007/11510888_26.
[189] A. Wang, Y. Li, and W. Wang, “Text Clustering Based on Key Phrases,” in *2009 First International Conference on Information Science and Engineering*, Dec. 2009, pp. 986–989. doi: 10.1109/ICISE.2009.1163.
[190] A. Skabar and K. Abdalgader, “Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 62–75, Jan. 2013, doi: 10.1109/TKDE.2011.205.
[191] L. Jing, M. K. Ng, and J. Z. Huang, “Knowledge-based vector space model for text clustering,” *Knowl. Inf. Syst.*, vol. 25, no. 1, pp. 35–55, Oct. 2010, doi: 10.1007/s10115-009-0256-5.
[192] S. Zhong and J. Ghosh, “A Comparative Study of Generative Models for Document Clustering,” *SIAM Knowl. Inf. Syst.*, 2002.
[193] Y. Zhao and G. Karypis, “Comparison of Agglomerative and Partitional Document Clustering Algorithms:,” Defense Technical Information Center, Fort Belvoir, VA, Apr. 2002. doi: 10.21236/ADA439503.
[194] L. Jing, M. K. Ng, J. Xu, and J. Z. Huang, “Subspace Clustering of Text Documents with Feature Weighting K-Means Algorithm,” in *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg, 2005, pp. 802–812. doi: 10.1007/11430919_94.
[195] M. Steinbach, G. Karypis, and V. Kumar, “A Comparison of Document Clustering Techniques,” Report, May 2000. Accessed: Jan. 31, 2022. [Online]. Available: http://conservancy.umn.edu/handle/11299/215421
[196] B. C. M. Fung, K. Wang, and M. Ester, “Hierarchical Document Clustering Using Frequent Itemsets,” in *Proceedings of the 2003 SIAM International Conference on Data Mining*, May 2003, pp. 59–70. doi: 10.1137/1.9781611972733.6.
[197] Y. Zhao, G. Karypis, and U. Fayyad, “Hierarchical Clustering Algorithms for Document Datasets,” *Data Min. Knowl. Discov.*, vol. 10, no. 2, pp. 141–168, Mar. 2005, doi: 10.1007/s10618-005-0361-3.
[198] R. Gil-García and A. Pons-Porrata, “Dynamic hierarchical algorithms for document clustering,” *Pattern Recognit. Lett.*, vol. 31, no. 6, pp. 469–477, Apr. 2010, doi: 10.1016/j.patrec.2009.11.011.
[199] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, “Incremental hierarchical clustering of text documents,” in *Proceedings of the 15th ACM international conference on Information and knowledge management*, New York, NY, USA, Nov. 2006, pp. 357–366. doi: 10.1145/1183614.1183667.
[200] C. C. Aggarwal and P. S. Yu, “A Framework for Clustering Massive Text and Categorical Data Streams,” in *Proceedings of the 2006 SIAM International Conference on Data Mining*, Apr. 2006, pp. 479–483. doi: 10.1137/1.9781611972764.44.
[201] Y.-B. Liu, J.-R. Cai, J. Yin, and A. W.-C. Fu, “Clustering Text Data Streams,” *J. Comput. Sci. Technol.*, vol. 23, no. 1, pp. 112–128, Jan. 2008, doi: 10.1007/s11390-008-9115-1.
[202] S. Zhong, “Efficient streaming text clustering,” *Neural Netw.*, vol. 18, no. 5, pp. 790–798, Jul. 2005, doi: 10.1016/j.neunet.2005.06.008.
[203] S. Yang, G. Huang, and B. Cai, “Discovering Topic Representative Terms for Short Text Clustering,” *IEEE Access*, vol. 7, pp. 92037–92047, 2019, doi: 10.1109/ACCESS.2019.2927345.
[204] M. L. Errecalde, D. A. Ingaramo, and P. Rosso, “A new AntTree-based algorithm for clustering short-text corpora,” *J. Comput. Sci. Technol.*, vol. 10, no. 1, Apr. 2010, Accessed: Feb. 02, 2022. [Online]. Available: http://sedici.unlp.edu.ar/handle/10915/9660
[205] C. T. Zheng, C. Liu, and H. S. Wong, “Corpus-based topic diffusion for short text clustering,” *Neurocomputing*, vol. 275, pp. 2444–2458, Jan. 2018, doi: 10.1016/j.neucom.2017.11.019.
[206] X. Ni, X. Quan, Z. Lu, L. Wenyin, and B. Hua, “Short text clustering by finding core terms,” *Knowl. Inf. Syst.*, vol. 27, no. 3, pp. 345–365, Jun. 2011, doi: 10.1007/s10115-010-0299-7.
[207] J. Yin and J. Wang, “A dirichlet multinomial mixture model-based approach for short text clustering,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Aug. 2014, pp. 233–242. doi: 10.1145/2623330.2623715.
[208] W. Song and S. C. Park, “Genetic algorithm for text clustering based on latent semantic indexing,” *Comput. Math. Appl.*, vol. 57, no. 11, pp. 1901–1907, Jun. 2009, doi: 10.1016/j.camwa.2008.10.010.

[209] X. Cui, T. E. Potok, and P. Palathingal, “Document clustering using particle swarm optimization,” in *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, Jun. 2005, pp. 185–191. doi: 10.1109/SIS.2005.1501621.
[210] R. Janani and S. Vijayarani, “Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization,” *Expert Syst. Appl.*, vol. 134, pp. 192–200, Nov. 2019, doi: 10.1016/j.eswa.2019.05.030.
[211] L. M. Abualigah, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, “A krill herd algorithm for efficient text documents clustering,” in *2016 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, May 2016, pp. 67–72. doi: 10.1109/ISCAIE.2016.7575039.
[212] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis,” *Eng. Appl. Artif. Intell.*, vol. 73, pp. 111–125, Aug. 2018, doi: 10.1016/j.engappai.2018.05.003.
[213] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “Hybrid clustering analysis using improved krill herd algorithm,” *Appl. Intell.*, vol. 48, no. 11, pp. 4047–4071, Nov. 2018, doi: 10.1007/s10489-018-1190-6.
[214] A. Rangrej, S. Kulkarni, and A. V. Tendulkar, “Comparative study of clustering techniques for short text documents,” in *Proceedings of the 20th international conference companion on World wide web*, New York, NY, USA, Mar. 2011, pp. 111–112. doi: 10.1145/1963192.1963249.
[215] S. Seifzadeh, A. K. Farahat, M. S. Kamel, and F. Karray, “Short-Text Clustering using Statistical Semantics,” in *Proceedings of the 24th International Conference on World Wide Web*, New York, NY, USA, May 2015, pp. 805–810. doi: 10.1145/2740908.2742474.
[216] S. Banerjee, K. Ramanathan, and A. Gupta, “Clustering short texts using wikipedia,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, Jul. 2007, pp. 787–788. doi: 10.1145/1277741.1277909.
[217] C. Jia, M. B. Carson, X. Wang, and J. Yu, “Concept decompositions for short text clustering by identifying word communities,” *Pattern Recognit.*, vol. 76, pp. 691–703, Apr. 2018, doi: 10.1016/j.patcog.2017.09.045.
[218] A. Huang, “Similarity Measures for Text Document Clustering,” p. 9.
[219] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, “A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture,” *IEEE Access*, vol. 6, pp. 39501–39514, 2018, doi: 10.1109/ACCESS.2018.2855437.
[220] T. Jo, “NTSO (Neural Text Self Organizer): A New Neural Network for Text Clustering,” vol. 1, no. 1, p. 13, 2010.
[221] J. Yi, Y. Zhang, X. Zhao, and J. Wan, “A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network,” *Math. Probl. Eng.*, vol. 2017, p. e8310934, Mar. 2017, doi: 10.1155/2017/8310934.
[222] J. Xu *et al.*, “Short Text Clustering via Convolutional Neural Networks,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, Jun. 2015, pp. 62–69. doi: 10.3115/v1/W15-1509.
[223] A. Hadifar, L. Sterckx, T. Demeester, and C. Develder, “A Self-Training Approach for Short Text Clustering,” in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy, 2019, pp. 194–199. doi: 10.18653/v1/W19-4322.
[224] J. Xu *et al.*, “Self-Taught Convolutional Neural Networks for Short Text Clustering,” *Neural Netw.*, vol. 88, pp. 22–31, Apr. 2017, doi: 10.1016/j.neunet.2016.12.008.
[225] B. Wang, W. Liu, Z. Lin, X. Hu, J. Wei, and C. Liu, “Text clustering algorithm based on deep representation learning,” *J. Eng.*, vol. 2018, no. 16, pp. 1407–1414, 2018, doi: 10.1049/joe.2018.8282.
[226] Y. Fan, L. Gongshen, M. Kui, and S. Zhaoying, “Neural Feedback Text Clustering With BiLSTM-CNN-Kmeans,” *IEEE Access*, vol. 6, pp. 57460–57469, 2018, doi: 10.1109/ACCESS.2018.2873327.
[227] W. Zhang, C. Dong, J. Yin, and J. Wang, “Attentive Representation Learning with Adversarial Training for Short Text Clustering,” *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3052244.
[228] “Use of artificial intelligence in infectious diseases - ScienceDirect.” https://www.sciencedirect.com/science/article/pii/B9780128171332000185 (accessed Feb. 07, 2022).
[229] R. Xu and D. C. Wunsch, “Clustering Algorithms in Biomedical Research: A Review,” *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 120–154, 2010, doi: 10.1109/RBME.2010.2083647.

[230] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, “A roadmap of clustering algorithms: finding a match for a biomedical application,” *Brief. Bioinform.*, vol. 10, no. 3, pp. 297–314, May 2009, doi: 10.1093/bib/bbn058.
[231] “Artificial intelligence in healthcare: past, present and future | Stroke and Vascular Neurology.” https://svn.bmj.com/content/2/4/230.abstract (accessed Feb. 07, 2022).
[232] O. Gencoglu, “Deep Representation Learning for Clustering of Health Tweets,” *ArXiv190100439 Cs Stat*, Dec. 2018, Accessed: Aug. 16, 2021. [Online]. Available: http://arxiv.org/abs/1901.00439
[233] M. R. Karim *et al.*, “Deep learning-based clustering approaches for bioinformatics,” *Brief. Bioinform.*, vol. 22, no. 1, pp. 393–415, Jan. 2021, doi: 10.1093/bib/bbz170.
[234] “Fuzzy Approach Topic Discovery in Health and Medical Corpora | SpringerLink.” https://link.springer.com/article/10.1007/s40815-017-0327-9 (accessed Feb. 07, 2022).
[235] Y. Lu, P. Zhang, J. Liu, J. Li, and S. Deng, “Health-Related Hot Topic Detection in Online Communities Using Text Clustering,” *PLOS ONE*, vol. 8, no. 2, p. e56221, Feb. 2013, doi: 10.1371/journal.pone.0056221.
[236] B. Lwowski and P. Najafirad, “COVID-19 Surveillance through Twitter using Self-Supervised Learning and Few Shot Learning,” Jun. 2020, Accessed: Feb. 07, 2022. [Online]. Available: https://openreview.net/forum?id=HnTelVEqX-l
[237] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, “Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—A study to show how popularity is affecting accuracy in social media,” *Appl. Soft Comput.*, vol. 97, p. 106754, Dec. 2020, doi: 10.1016/j.asoc.2020.106754.
[238] L. L. Wang and K. Lo, “Text mining approaches for dealing with the rapidly expanding literature on COVID-19,” *Brief. Bioinform.*, vol. 22, no. 2, pp. 781–799, Mar. 2021, doi: 10.1093/bib/bbaa296.
[239] “Social Media Mining with Dynamic Clustering: A Case Study by COVID-19 Tweets | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/abstract/document/9319496 (accessed Feb. 07, 2022).
[240] M. Asgari-Chenaghlu, N. Nikzad-Khasmakhi, and S. Minaee, “Covid-Transformer: Detecting COVID-19 Trending Topics on Twitter Using Universal Sentence Encoder,” *ArXiv200903947 Cs*, Sep. 2020, Accessed: Feb. 07, 2022. [Online]. Available: http://arxiv.org/abs/2009.03947
[241] A. Chakraborty and S. Bose, “Around the world in 60 days: an exploratory study of impact of COVID-19 on online global news sentiment,” *J. Comput. Soc. Sci.*, vol. 3, no. 2, pp. 367–400, Nov. 2020, doi: 10.1007/s42001-020-00088-3.
[242] S. Praveen, R. Ittamalla, and G. Deepak, “Analyzing the attitude of Indian citizens towards COVID-19 vaccine – A text analytics study,” *Diabetes Metab. Syndr. Clin. Res. Rev.*, vol. 15, no. 2, pp. 595–599, Mar. 2021, doi: 10.1016/j.dsx.2021.02.031.
[243] S. W. H. Kwok, S. K. Vadde, and G. Wang, “Tweet Topics and Sentiments Relating to COVID-19 Vaccination Among Australian Twitter Users: Machine Learning Analysis,” *J. Med. Internet Res.*, vol. 23, no. 5, p. e26953, May 2021, doi: 10.2196/26953.
[244] E. E. Milios, M. M. Shafiei, S. Wang, R. Zhang, B. Tang, and J. Tougas, “A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering A Systematic Study on Document Representation and Dimensionality Reduction for Text Clustering.”
[245] X. Hu and I. Yoo, “A comprehensive comparison study of document clustering for a biomedical digital library MEDLINE,” in *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL ’06)*, Jun. 2006, pp. 220–229. doi: 10.1145/1141753.1141802.
[246] “Document Clustering: The Next Frontier | David C. Anastasiu, Andrea Ta.” https://www.taylorfrancis.com/chapters/edit/10.1201/9781315373515-13/document-clustering-next-frontier-david-anastasiu-andrea-tagarelli-george-karypis (accessed Aug. 18, 2021).
[247] V. Vinay, I. J. Cox, K. Wood, and N. Milic-Frayling, “A comparison of dimensionality reduction techniques for text retrieval,” in *Fourth International Conference on Machine Learning and Applications (ICMLA’05)*, Dec. 2005, p. 6 pp.-. doi: 10.1109/ICMLA.2005.2.
[248] A. Kumar, “Analysis of unsupervised dimensionality reduction techniques,” *Comput. Sci. Inf. Syst.*, vol. 6, no. 2, pp. 217–227, 2009, doi: 10.2298/CSIS0902217K.
[249] S. V. S and S. Surendran, “A Review of Various Linear and Non Linear Dimensionality Reduction Techniques.”
[250] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” *ArXiv180203426 Cs Stat*, Sep. 2020, Accessed: Feb. 16, 2022. [Online]. Available: http://arxiv.org/abs/1802.03426

[251] “A Survey on Text Representation and Similarity Calculation in Text Clustering--《Information Science》2012年04期.” https://en.cnki.com.cn/Article_en/CJFDTotal-QBKX201204031.htm (accessed Aug. 18, 2021).
[252] M. Korzycki, I. Gatkowska, and W. Lubaszewski, “2 - Can the Human Association Norm Evaluate Machine-Made Association Lists?,” in *Cognitive Approach to Natural Language Processing*, B. Sharp, F. Sèdes, and W. Lubaszewski, Eds. Elsevier, 2017, pp. 21–40. doi: 10.1016/B978-1-78548-253-3.50002-0.
[253] M. Grootendorst and N. Reimers, “MaartenGr/BERTopic: v0.9.4.” Zenodo, Dec. 14, 2021. doi: 10.5281/zenodo.5779238.
[254] “2.3. Clustering,” *scikit-learn*. https://scikit-learn/stable/modules/clustering.html (accessed Apr. 28, 2022).
[255] “Exploring the Space of Topic Coherence Measures | Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.” https://dl.acm.org/doi/10.1145/2684822.2685324 (accessed Apr. 28, 2022).
[256] P. Dahal, “Deep Clustering,” *DeepNotes*, Apr. 24, 2019. http://deepnotes.io/deep-clustering (accessed Aug. 15, 2021).
[257] U. David and M. Karabatak, “Text Clustering of COVID-19 Vaccine Tweets,” in *2022 10th International Symposium on Digital Forensics and Security (ISDFS)*, Jun. 2022, pp. 1–6. doi: 10.1109/ISDFS55398.2022.9800754.
[258] D. Murthy, “Twitter: Microphone for the masses?,” *Media Cult. Soc.*, vol. 33, no. 5, pp. 779–789, Jul. 2011, doi: 10.1177/0163443711404744.
[259] A. A. Aslam *et al.*, “The Reliability of Tweets as a Supplementary Method of Seasonal Influenza Surveillance,” *J. Med. Internet Res.*, vol. 16, no. 11, p. e3532, Nov. 2014, doi: 10.2196/jmir.3532.
[260] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, “Detecting Spammers on Twitter,” p. 10.
[261] B. Huang and K. M. Carley, “Disinformation and Misinformation on Twitter during the Novel Coronavirus Outbreak,” *ArXiv200604278 Cs*, Jun. 2020, Accessed: Apr. 18, 2022. [Online]. Available: http://arxiv.org/abs/2006.04278
[262] Dav-web, “MSC.-Thesis.” Mar. 14, 2022. Accessed: Apr. 29, 2022. [Online]. Available: https://github.com/Dav-web/MSC.-Thesis

APPENDICES

APPENDIX- 1: CODEBASE FOR TEXT CLUSTERING AND TOPIC MODELING OF COVID-19 VACCINE DATASET

All interactive plots are available for interaction on GitHub at https://github.com/Dav-web/MSC.-Thesis [262]