An Unsupervised Approach To COVID-19 Fake Tweet Detection

A thesis submitted in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE

Department of Statistical Sciences Faculty of Science

UNIVERSITY OF CAPE TOWN

By: Bulungisa Jarana

Supervisor: Mzabalazo Ngwenya

Abstract

Context: With the ongoing COVID-19 pandemic, social media platforms have become a crucial source of information. However, not all information shared on these platforms is accurate. The dissemination of fake news, intentional or unintentional, can lead to panic among readers and further exacerbate the effects of the pandemic.
Objectives: This research project aims to explore the potential of unsupervised machine learning algorithms in differentiating between genuine and fake COVID-19 news shared on Twitter. The methodology includes a literature review, experimental analysis, and the utilization of Twitter data.
Methods: The study used Mini-Batch K-means and K-means clustering algorithms to provide us with 'grouping' of Twitter data into the two clusters. Word embedding techniques such as TF-IDF, Word2Vec, and BERT were employed to generate features for the unsupervised machine learning models.
Results: The results on test data show that K-means algorithm was the best performing algorithm (76% accuracy was achieved) in determining fake tweets about Covid-19. K-means algorithm using Bert word embedding is the best performing model followed by Mini-Batch K-means using TF-IDF word embedding (69% accuracy was achieved).
Conclusions: The study demonstrates that clustering Twitter COVID-19 news as genuine or fake using K-means and Mini-Batch K-means algorithms is feasible.

Keywords: Clustering, Machine Learning, unsupervised learning, K-Means, Mini-Batch K-Means, TF-IDF, Word2Vec, Bert, Truncated SVD (Singular Value Decomposition), t-distributed stochastic neighbourhood embedding (t-SNE)

Acknowledgments

I would like to express my deepest gratitude to the individuals who have played a pivotal role in the completion of this master's thesis.

Firstly, I want to dedicate this work to my beloved mother, maMthembu (Nozy), who, despite her untimely departure during the course of my master's degree, remains a source of inspiration. Her unwavering belief in my abilities echoes in every word of this thesis. I am forever indebted to her for instilling in me the importance of education and perseverance.

Heartfelt thanks to my father and siblings for their boundless love and encouragement. Their unwavering support provided the emotional foundation that allowed me to navigate the challenges of academia. Their belief in my abilities has been a motivating force, and I am grateful for their enduring presence in my life.

To my esteemed friend Mno, I express profound gratitude for your invaluable assistance.

I am also profoundly thankful to my loving wife, Milisa, whose support and understanding have been my pillars of strength throughout this challenging journey. Her encouragement and sacrifices have fuelled my determination to overcome obstacles and reach this academic milestone.

A special acknowledgment to my children, Mhlaba and ZanoMhlaba, whose laughter and presence brought joy to the challenging moments of this journey. Your resilience and curiosity inspire me to strive for excellence, not just for myself but for the legacy we are building together.

A special mention is due to my supervisor, Mzabalazo, whose guidance, patience, and unwavering support have been instrumental in shaping this thesis. Your insightful feedback and dedication to my academic growth have made a lasting impact on my intellectual development.

Lastly, I extend my thanks to myself. It is crucial to recognize and appreciate the personal dedication, resilience, and self-belief that have been the driving forces behind this achievement. I am proud to have stayed committed to my goals and to have believed in myself even during the most demanding moments.

This thesis is more than a scholarly accomplishment; it is a tribute to those who have touched my life in profound ways. Their influence will resonate not only in these pages but in my continued pursuit of knowledge and personal growth.

With heartfelt gratitude,
Bulungisa Jarana

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 BACKGROUND AND INTRODUCTION

1. INTRODUCTION

This study explored the issue of detecting fake news on Twitter during the COVID-19 pandemic. The focus was on developing a machine learning model that can effectively identify fake news using people's tweets shared on Twitter's social media platform. We employed unsupervised learning classification to train and test the Twitter datasets collected. Several different machine learning algorithms were used by researcher to cluster Twitter Covid-19 news as either fake or genuine.

1.1 BACKGROUND TO THE STUDY

On December 31, 2019, the China Country Office of the World Health Organization (WHO) received reports regarding instances of pneumonia of unknown origin in Wuhan City, situated in the Hubei Province of China (World Health Organization, 2020). The World Health Organization received notifications from national authorities in China regarding a cohort of 44 individuals afflicted with pneumonia of indeterminate origin. Most of these patients (33) were in stable condition, while the remaining 11 were severely ill (World Health Organization, 2020). The symptoms observed in these patients are similar to those of various respiratory illnesses. Due to a surge in case notifications from various regions in China and across the globe, the Emergency Committee of the World Health Organization declared a worldwide public health emergency on January 30, 2020. The origin of the current outbreak of the novel coronavirus SARS-CoV-2 (previously known as 2019-nCoV) can be traced back to this point (Velavan and Meyer, 2020). According to Velavan and Meyer (2020), coronaviruses are large, positive single-stranded RNA viruses that have an envelope and can infect not only humans, but also a diverse array of animals. Tyrell and Bynoe (1967) were the first to report coronaviruses in 1966 after isolating the viruses from individuals with the common cold. The viruses were named "coronaviruses" due to their spherical shape and the presence of surface projections that resemble a solar corona, giving them the appearance of a crown (Latin: corona = crown) (Tyrrell and Bynoe, 1967).

While the majority of the patients had mild condition, the disease had proven to be fatal. Patients suffering from various diseases continue to die in increasing numbers across the globe. As of 5:18pm CET on March 4, 2022, there have been 440,807,756 verified cases of COVID-19 worldwide, resulting in 5,978,096 fatalities, reported to World Health Organisation (WHO), (https://covid19.who.int/, 2022).

Figure 1.1 below shows that the number of confirmed cases has been decreasing in recent months. Figure 1.2 shows the distribution of confirmed cases by region. COVID-19 is affecting many countries worldwide, with Europe being the hardest hit, having over 181 million confirmed cases. America follows with 147 million confirmed cases, while Africa has the lowest number of confirmed cases at 8 million. Despite its record of having the highest tally of confirmed cases, Europe assumes the secondary position with regard to the COVID-19 mortality count. The Americans region has the highest number of deaths as seen in Figure 1.3. Africa has the lowest number of deaths, with approximately 170,000 deaths.

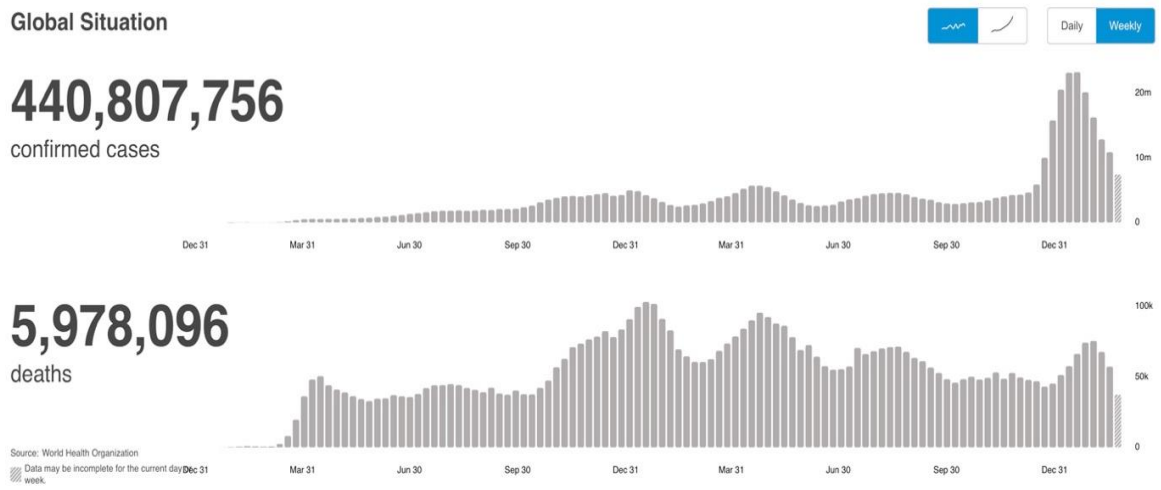

Figure 1.1: Covid confirmed cases and death
source: WHO Coronavirus (COVID-19) Dashboard (https://covid19.who.int/, 2022).

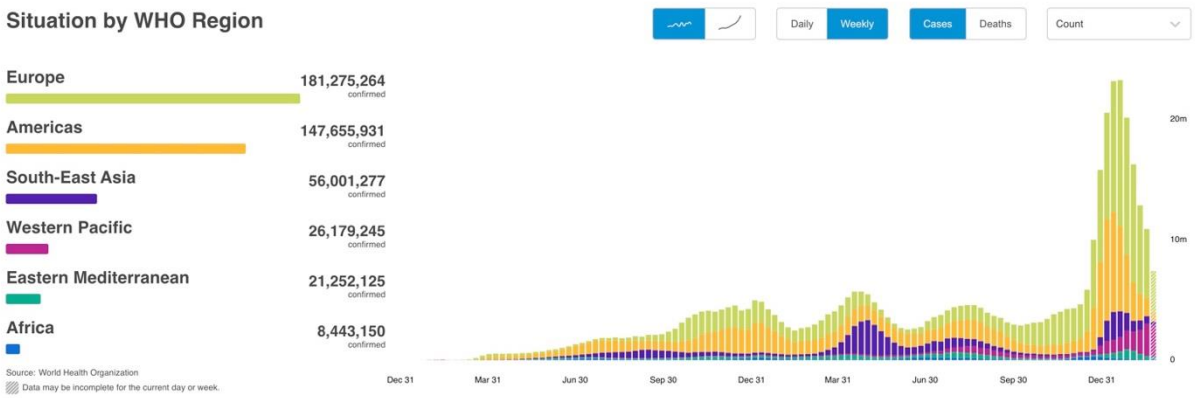

Figure 1.2: Covid confirmed by Region
source: WHO Coronavirus (COVID-19) Dashboard (https://covid19.who.int/, 2022).

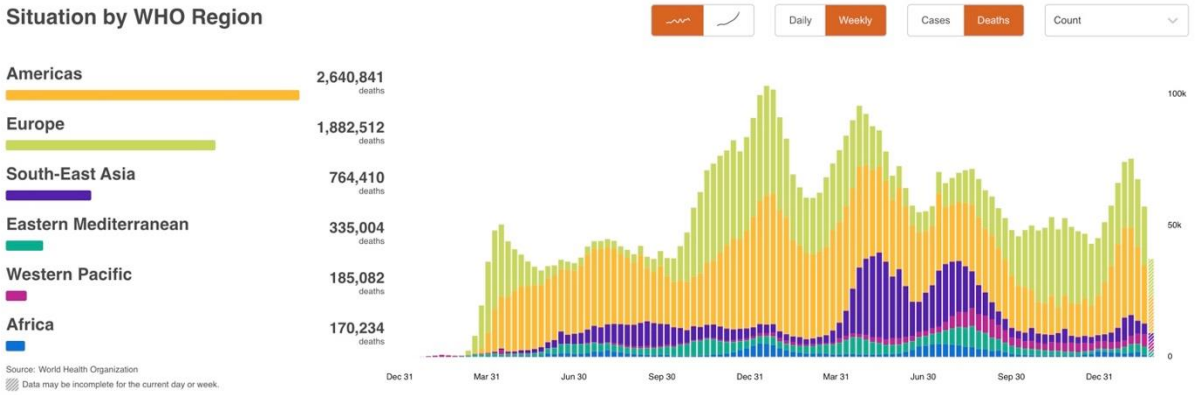

Figure 1.3: Covid deaths by Region
source: WHO Coronavirus (COVID-19) Dashboard (https://covid19.who.int/, 2022).

According to the information provided to the WHO, between January 3, 2020, and March 23, 2022, South Africa documented 3,705,696 confirmed cases of COVID-19, which resulted in 99,893 deaths, (https://covid19.who.int/, 2022).

The graphical representation in Figure 1.4 depicts the cumulative count of confirmed cases and fatalities across the four distinct waves of the pandemic.

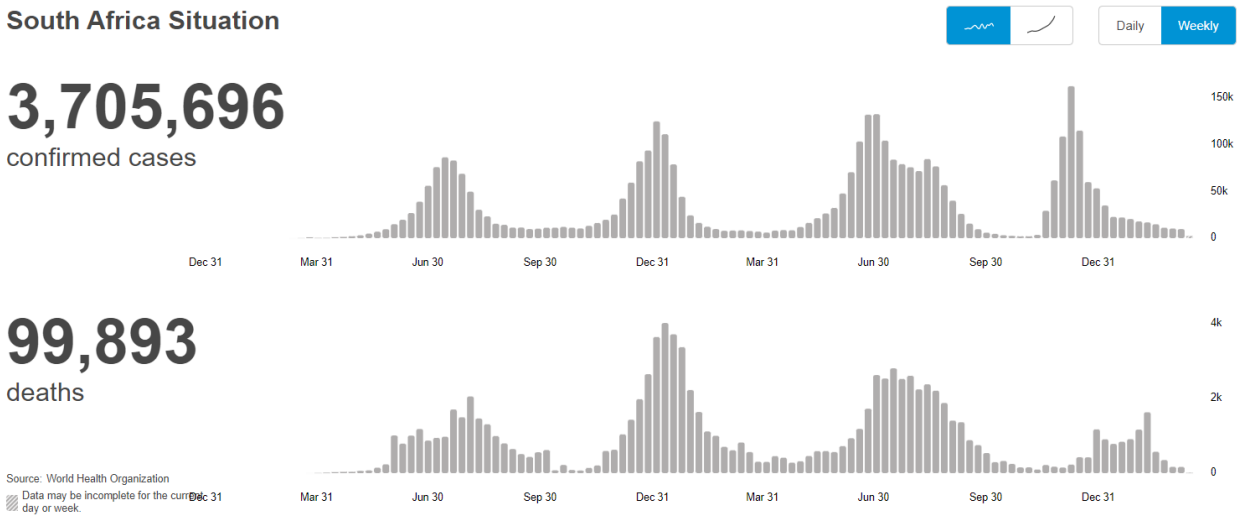

Figure 1.4: Covid confirmed cases and death in RSA
source: WHO Coronavirus (COVID-19) Dashboard (https://covid19.who.int/, 2022).

The prevalence of COVID-19-related discussions on social media can be attributed to the staggering number of people affected by the pandemic. However, not all information shared on these platforms is accurate. Unintentionally or otherwise, individuals tend to share fake news, which leads to widespread panic and the dissemination of misinformation, aggravating the ramifications of the pandemic. To mitigate the dissemination of fabricated information and misleading content, governments and health authorities worldwide have been engaged in efforts to educate citizens on how to minimize their exposure to the virus, while also combating false information circulating on public and private social networks (UNDP.org, 2020). In accordance with the findings of Alcott and Gentzkow (2017), the term "fake news" encompasses deliberately fabricated information with the potential to misinform and deceive its readership.

1.2 PROBLEM STATEMENT

According to existing literature, misinformation regarding COVID-19 has proliferated on social media, encompassing a broad range of false claims such as the promoting fake remedies "cures" like gargling with salt or lemon water or injecting oneself with bleach (Biral, 2021). Additionally, unfounded conspiracy theories have been circulating on social media, such as the notion that the virus was created in a laboratory in Wuhan (Andersen et al., 2020; Cohen, 2020), and baseless claims that the 5G cellular network is responsible for causing or worsening COVID-19 symptoms (BBC News, 2020). Prominent political leaders, including former American President Trump and Brazilian President Jair Bolsonaro, have also contributed to the spread of COVID-19 misinformation by endorsing unproven treatments such as hydroxychloroquine, claiming that it works in all cases (Constine, 2020). The Office of Communications (Ofcom), is the government-approved regulatory and competition authority for the broadcasting, conducted a survey in the United Kingdom and found that 50% of the population reported being exposed to false information regarding the coronavirus (Ofcom, 2020). Pew Research Centre has also reported comparable findings in the United States, with 48% of respondents indicating exposure to COVID-19 misinformation (Mitchell and Oliphant, 2020). Of those who reported exposure to false information, almost two-thirds (66%) claimed to have encountered it daily, which poses a concern as repeated exposure has been shown to strengthen belief in misinformation (Pennycook et al., 2018). Classification of any news into fake news has generated great interest from researchers around the globe. Previous studies on identifying fake news have utilized various conventional machine learning techniques and neural networks. However, their emphasis has been primarily on identifying political news pieces and similar types of content (Wang, 2017). Using already existing models would likely be negatively affected by dataset bias and their effectiveness in identifying news on a different topic may be limited.

The above raises the following research question:

• Can a distinction be automatically made between fake news and authentic news pertaining to COVID-19 using machine learning approaches?

1.3 AIMS & OBJECTIVE OF THE STUDY

The aim of this thesis is not to identify a single universally optimal text clustering technique, as it is not feasible to do so. The experiments will focus solely on a particular type of text data, (Twitter data). This thesis will investigate a limited number of methods for gauging similarity between texts and clustering them based on these similarities. It is important to note that there are countless other ways to accomplish this task, but it is beyond the scope of this thesis to evaluate all of them. As the text data in our context is not labelled, an unsupervised approach is mandatory.

Primary aim:

- The primary aim of this study is to develop an unsupervised model to classify Covid-19 news related Tweets as fake or real.

The research objectives are as follows:

1. Identify the best NLP technique out all the methods considered for use in processing Twitter text in the context of COVID-19 fake news identification.
2. Identify the best unsupervised clustering method between K-means and Mini-Batch K-means for detection of COVID-19 fake news from twitter data.
3. Determine the best combination of NLP method and unsupervised classifier to use in the detection of COVID-19 fake news in tweets.

1.4 OUTLINE OF THESIS

The study follows a structured layout, comprising distinct chapters that contribute to the overall research framework. The layout of the study is outlined as follows:

- Chapter 1: Study Scope and Nature: This chapter presents the rationale for conducting the study, along with the problem statement and the primary objectives of the research.
- Chapter 2: Literature Review: Within this chapter, a comprehensive examination of the existing literature on the classification of fake news is provided. It encompasses an extensive review of related works and a critical discussion of relevant studies conducted in the field.

- Chapter 3: Research methodology. This chapter offers an in-depth account of the research methodology employed to collect and analyse the data. It outlines the research design, data collection methods, and data analysis techniques.
- Chapter 4: Experiments. This chapter describes the experiments conducted in detail. It explains the system's features, models, and how the collected data was analysed. Additionally, the chapter interprets the results derived from the experiments conducted.
- Chapter 5: Conclusions and Future Work: This concluding chapter synthesizes the findings presented in Chapter 4. Additionally, it discusses future directions for research and potential areas of further investigation.

The next chapter discusses the literature review on clustering news to either fake or legitimate.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

In Chapter 1, the rationale for conducting the study was presented. This chapter will define what fake news is and explore the use of both supervised and unsupervised learning algorithms to detect COVID-19 fake or real news. The literature review will serve as the basis for the researcher's experiments.

2.2 DEFINITION OF FAKE NEWS

The Macmillan English Dictionary defines fake news as "a story that is presented as being a genuine item of news but is in fact not true and is intended to deceive people." Conroy, Rubin, and Chen (2015) argued that three distinct categories of fake news exist, namely: Type A, which comprises serious fabrications, Type B, which encompasses large-scale hoaxes, and Type C, which includes humorous fakes. Allcott and Gentzkow (2016) have a much narrower definition of fake news; they defined it as “a news article that is intentionally and verifiably false and could mislead readers.” This definition aids in eliminating any uncertainty between fake news and other associated notions, such as hoaxes and satires. As explicated by Duffy et al. (2019), the conceptualization of fake news involves the creation of spurious content that mimics authentic news sources, employing subtle strategies to ensnare the public into perceiving it as genuine. In accordance with the definition put forth by Rubin et al. (2016), fake news can be defined as misleading information that encompasses fabrications, satirical content, hoaxes, and other forms of false information.

In this thesis, we propose the following definition for fake news: "Fake news refers to any news that misrepresents the truth." By including all types of news, this definition will encompass fake tweets, which will be discussed and covered in this study. For a tweet to be considered true, all the text within the tweet must be truthful. This narrow definition will greatly assist in clustering the tweets in the following chapters.

Figure 2.1 displays fake news and its related components, which can be considered the scope and diversity of online false information (Zhang and Ghorbani, 2020).

The onion-shaped diagram illustrates that the phrase "Fake News" is at the centre and comprises four primary constituents: Creator/Spreader, News Content, Target Victims, and Social Context. All these components occupy the first layer around "Fake News."

- Creator/Spreader: Individuals or automated bots (machines) can be responsible for producing fake news online.
- Target Victims: Fake news online is primarily aimed at victimizing unsuspecting targets.
- News Content: The body of a news article encompasses both tangible elements such as the headline, written text, videos, and images, as well as intangible components including the intended message, emotional tone, and subject matter.
- Social Context: Social context refers to the way news is disseminated across the internet. Analysis of the social context involves examining user network patterns and broadcast distribution patterns.

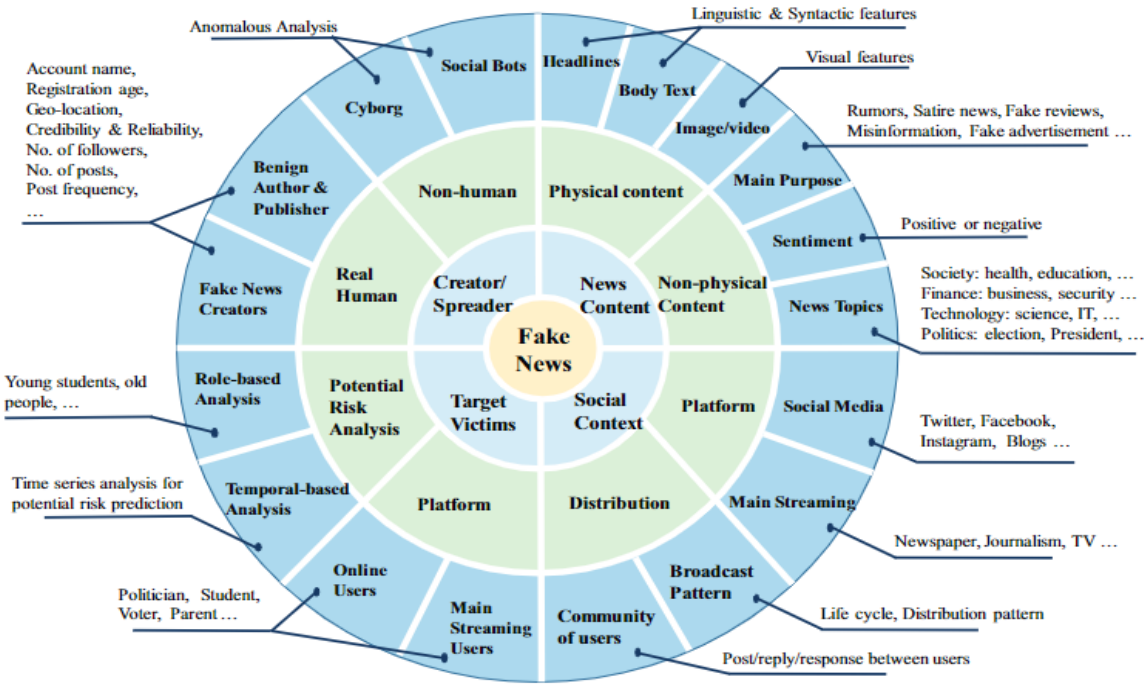

Figure 2.1: Anything connected to fake news. Source: Zhang and Ghorbani, (2020).

2.3 RELATED WORK

Granik and Mesyura (2017) presented a method for detecting fake news in their research, utilizing a naive Bayes classifier. Mitchell (1997) defined the naive Bayes classifier as a supervised classification technique that assumes the independence of variables given the classes. This classifier proves effective in learning situations where an example is defined by a combination of attributes, and the intended outcomes of the learning process can take on distinct and limited values. Granik and Mesyura (2017) applied and tested their method against a dataset of social media posts (Facebook news posts). They attained a classification accuracy of around 74%. The primary objective of their study was to assess the efficacy of this methodology in addressing the specific issue of detecting fake news using a news dataset that had been manually labelled. The objective was to determine whether artificial intelligence could be a viable solution for fake news detection.

The researchers of this study stated that there are several ways the accuracy can be improved:

- Try and increase the total size of the dataset used for training. They used a dataset that contained around 2000 articles, but after filtering a dataset only 1771 news articles were obtained.
- Including longer news articles in the dataset: The researchers suggest incorporating news articles that are substantially longer in length. The dataset used in the study primarily consisted of previews of longer articles.
- Remove stop words from the news articles. Granik and Mesyura (2017) did not eliminate stop words from the articles.
- Use stemming. Stemming involves the reduction of inflected words to their base, root, or stem form. Say for example you have three words send, sent and sending. All these words mean the same thing but are different tenses of the word send. This means after we stem the words, we will only have the word — send, for all three words.
- Treat uncommon words separately.
- Use a cluster of words to calculate probabilities, rather than individual words.

Wang (2017) adopted a different approach in the pursuit of detecting fake news. Wang (2017) employed the utilization of convolutional neural network (CNN) as a discerning methodology. This approach is known as Wang-CNN. The study used textual features and metadata to train different machine learning models. Wang (2017) utilized the LIAR dataset, which comprises of 12,836 brief statements categorized according to truthfulness, topic, context/setting, speaker, location, political affiliation, and past track record.

Table 2.1 shows the corpus statistics. The individuals speaking in the LIAR dataset consist of a combination of Democrats and Republicans.

Table 2.1: The LIAR dataset statistics. Source: Wang (2017).

Data Statistics	
Training set size	10269
Validation set size	1284
Testing set size	1283
Avg. statement length (tokens)	17.9
Top 3 Speaker Affiliations	
Democrats	4150
Republicans	5687
None (e.g., FB posts)	2185

The primary focus of Wang's (2017) study was centred on the utilization of a convolutional neural network (CNN). To capture the interdependence among metadata vectors, a convolutional layer was employed, followed by a bidirectional Long Short-Term Memory (LSTM) layer. The max-pooled text embeddings were combined with the metadata embeddings from the bidirectional LSTM. This composite representation was subsequently passed through a fully connected layer that employed a softmax activation function, resulting in the generation of the ultimate prediction.

Figure 2.2 depicts the architecture which employs a combination of Convolutional Neural Network and meta-data to form a hybrid framework for text integration.

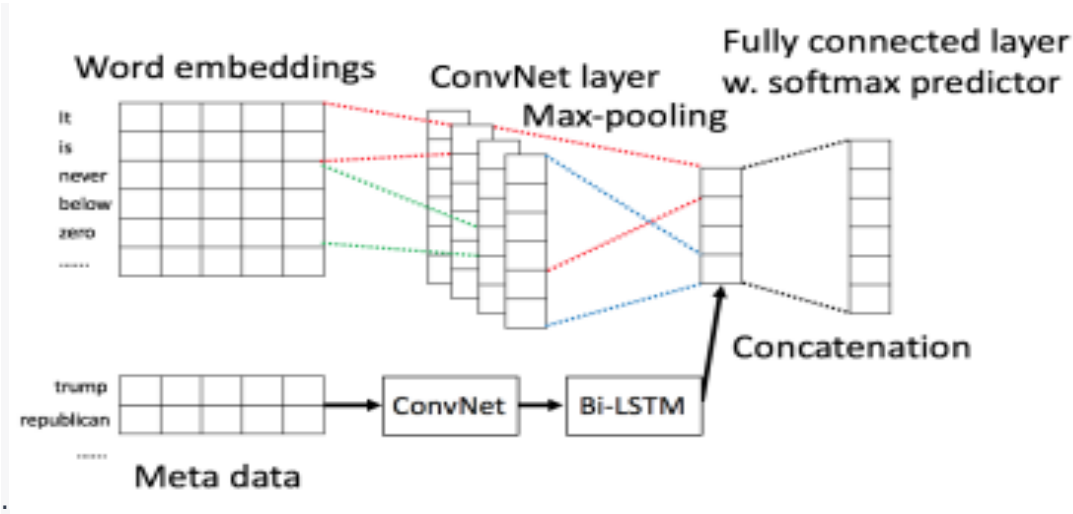

Figure 2.2: The framework for combining text and meta-data using a hybrid Convolutional Neural Network. Source: Wang (2017).

This research analysed a political domain dataset, which consisted of statements made by two parties, as well as metadata characteristics such as state, subject, speaker, party, context, job and history. Combining text and speaker features resulted in an accuracy of 27.7%, while combining all metadata elements with text achieved an accuracy of 27.4%, due to overfitting, the model did not perform well. Ahmad et al. (2020) investigated various textual features that can be used to discriminate between authentic and fabricated news. In their study thirteen distinct machine learning algorithms were subjected to training through diverse ensemble methods, followed by an assessment of their performance using four authentic real-world datasets. The initial dataset, known as the ISOT Fake News Dataset, was acquired from the World Wide Web. The subsequent two datasets were sourced from Kaggle and are publicly available. The fourth dataset was a combined dataset comprising articles from the three previous datasets. The ISOT Fake News Dataset encompassed a corpus of 44,898 articles, wherein 21,417 articles were classified as factual, and 23,481 articles were identified as fake. The first Kaggle dataset comprised 20,386 articles, while the second Kaggle dataset contained only 3,352 articles.

To measure the efficacy of models the researcher's used accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The present terminology comprises four key performance metrics namely, true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The researchers used the same approach as Wang (2017), employing bidirectional long short-term memory networks (Bi-LSTM). However, the team utilized distinct feature sets and referred to their approach as Wang-Bi-LSTM in their study. Additionally, they employed the linear SVM approach proposed by Pérez-Rosas et al. (2017) and referred to it as Perez-LSVM. Figure 2.3 exhibits the mean accuracy of the algorithms across the four datasets in the study by Ahmad et al (2020). Among the algorithms, the bagging classifier (decision trees) demonstrated the most notable performance, achieving an accuracy of 94% on the test data. Conversely, the Wang-Bi-LSTM (bidirectional long short-term memory networks) exhibited the poorest performance, attaining an accuracy of 64.25%. Furthermore, the highest accuracy achieved by an individual learner was recorded at 77.6%. In contrast, the ensemble learners yielded an accuracy of 92.25%.

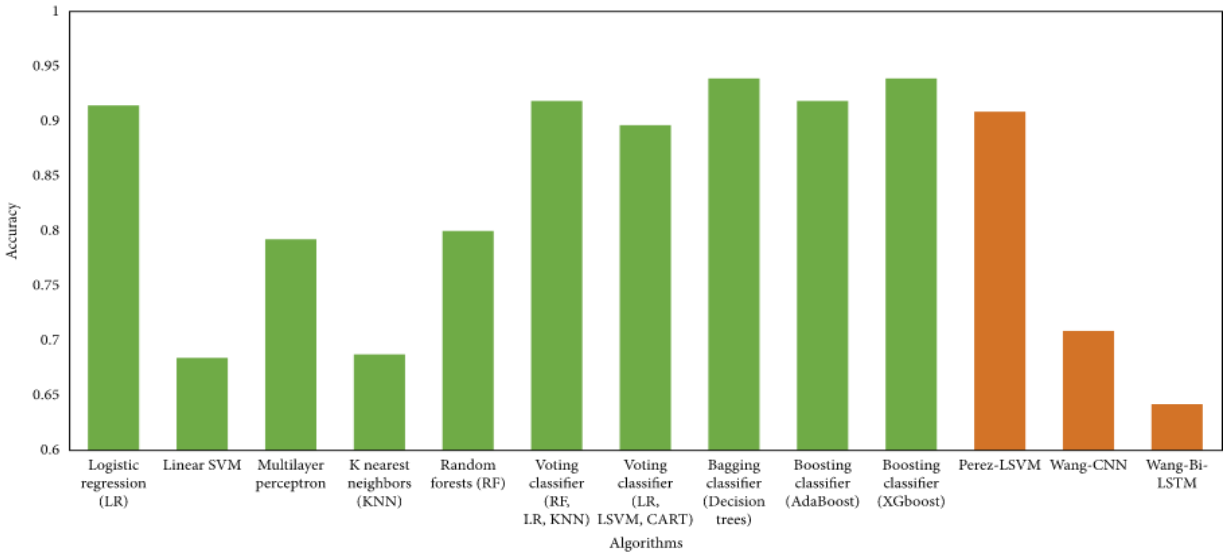

Figure 2.3: Average accuracy over all datasets. Source: Ahmad et al (2020).

The researchers also used the K-Nearest Neighbours (KNN) model, which harnesses a supervised machine learning approach that requires a dependent variable to make predictions on a given dataset.

The model was supplied with sufficient training data and tasked with determining the appropriate neighbourhood for each data point. In KNN models, the proximity of a new data point to its closest K neighbours is estimated, and the majority of its neighbours' votes determine its classification. If K equals 1, then the new data point is classified into the category with the nearest distance (Ahmad et al., 2020). Gangireddy et al. (2020) proposed using an unsupervised machine learning model for fake news detection. In their study, they devised GTUT (Graph mining methods over Textual, User, and Temporal data), a graph-based methodology that encompasses three distinct phases, serving as an approach to tackle this objective. They began by selecting a seed set of fake and genuine articles using broad observations on inter-user behaviour in the spread of fake news and subsequently proceeded to label all the articles in the dataset.

Figure 2.4 illustrates the stepwise labelling process employed by GTUT, wherein multiple criteria and methodologies are employed to assign comprehensive labels to entire articles within the dataset. As seen below GTUT uses a three-phase approach:

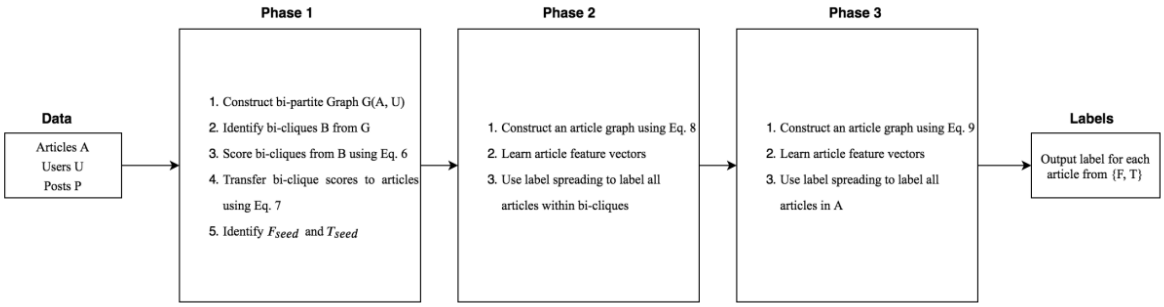

Figure 2.4: Block diagram of GTUT. Source: Gangireddy, Padmanabhan, Long and Chakraborty (2020).

During the preliminary stage, the researchers made high-level assumptions regarding the inter-user behavioural dynamics to identify a seed set of authentic and fabricated news articles. This was accomplished by identifying bi-cliques and evaluating their coherence based on textual and temporal factors. A bi-clique $B \in$ Ɓ can be defined as a combination of a set of users B_U ⊆ ᴜ and a set of articles B_A ⊆ A, where every user in B_U is connected to every article in B_A through an edge (Gangireddy et al, 2020), where articles and users are represented by A and U.

During the second phase, the labelling procedure is broadened to include all articles associated with bi-cliques, beyond the initial set of seeds. This was achieved through the utilization of three distinct forms of similarity information, including bi-clique similarity, user similarity, and textual similarity. Furthermore, the subsequent stage encompasses graph modelling, followed by graph embeddings and label spreading techniques. The focus of the third stage was to label the articles that are not part of the bi-cliques. This was accomplished by employing graph modelling and label spreading techniques, with the goal of assigning all articles in the dataset as either real or fake. Two datasets were used, PolitiFact and GossipCop Tables 2.2 and 2.3 shows the corpus statistics for these datasets.

Table 2.2: Statistics of the PolitiFact dataset. Source: Gangireddy et al (2020).

Types	Truthful/Real	Fake
Articles	369	367
Tweets	498005	355290
Users	283400	85208
Users posting both truthful and fake articles: 16060		

Table 2.3: Statistics of the GossipCop dataset. Source: Gangireddy et al (2020).

Types	Truthful/Real	Fake
Articles	600	450
Tweets	41580	123875
Users	6013	53288
Users posting both truthful and fake articles: 2520		

The accuracy values for the different models are presented in Table 2.4. As elucidated earlier, accuracy serves as a metric for assessing the efficacy of models. Based on the accuracy analysis, GTUT outperforms the baseline methods (UDF, TruthFinder & Majority Voting (MV)) with a significant margin. It accomplishes a notable improvement in accuracy of 10 percentage points across both datasets.

Table 2.4: Accuracy analysis. Source: Gangireddy et al (2020).

Method	PolitiFact	GossipCop
UDF	0.7	0.66
TruthFinder	0.59	0.67
MV	0.65	0.55
GTUT	0.8	0.77

Ahmed, Traore and Saad (2017) used text classification in detecting opinion spams and fabricated news. They explored the two primary categories of opinion spam, which are fabricated reviews and fabricated news. The common thread among the different forms of opinion spam is the presence of falsified content. Thus, they introduced a unified method that automatically detects fabricated content and can detect both counterfeit news and fabricated reviews. In their scholarly publication, the authors propose the adoption of a detection model that integrates various text analysis techniques, including n-gram features and term frequency metrics, in conjunction with machine learning classification methods. This approach would enable the identification of fabricated content, with a specific emphasis on counterfeit reviews and false news. Figure 2.5 shows how the researchers approached fake content detection. The first step involves pre-processing the dataset to eliminate redundant words and characters from the data.

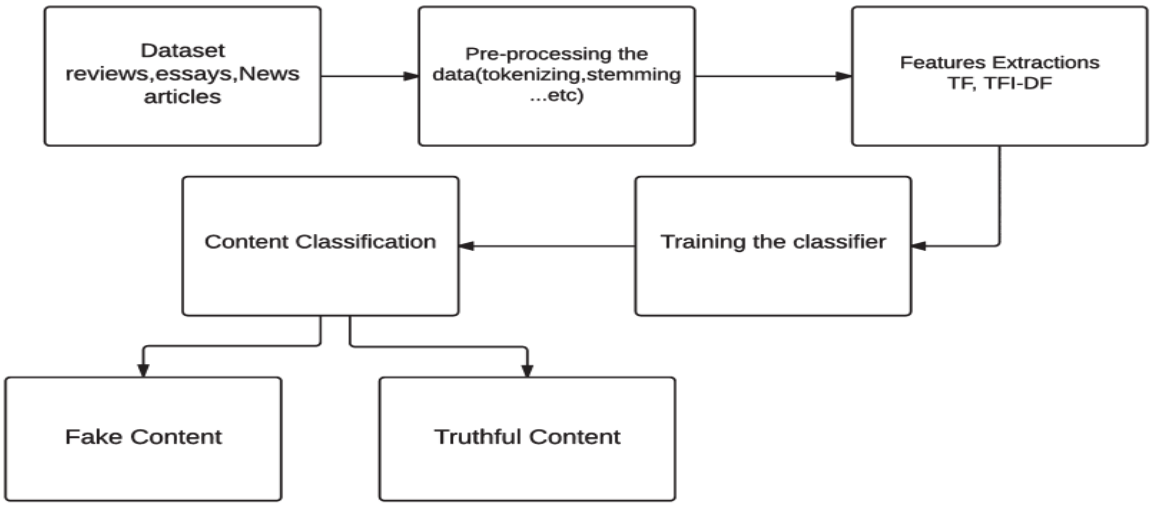

Figure 2.5: Classification process. Source: Ahmed, Traore and Saad (2017).

N-gram features are mined, and subsequently, a feature matrix is generated to show the documents under consideration. Subsequently, the classifier undergoes the final stage of the classification process, which involves training. The researchers examined several distinct machine learning algorithms in detail, which were SVM, linear support vector machines (LSVM), Logistic Regression (LR), decision trees (DT) and K-nearest neighbour (KNN). They also examined stochastic gradient descent (SGD) which is an optimization technique. Table 2.5 illustrates a comparison between the prior works and the findings of the researchers in terms of fake news and opinion spam detection models. As mentioned earlier, their methodology surpasses most of the current research. Through the implementation of their model on Ott et al.'s (2017) review dataset, the researchers attained a slightly superior outcome, achieving an accuracy of 90% in contrast to the previously obtained 89%. Moreover, they conducted supplementary experiments by applying their model to the news dataset compiled by Adali and Horne (2017). This dataset encompasses genuine news articles from BuzzFeed, along with satirical articles obtained from Burfoot and Baldwin's satire dataset. When distinguishing between fake and real news, they attained an accuracy of 87% by utilizing the LSVM algorithm in conjunction with n-gram features. As part of their study on fake news, the researchers gathered a fresh dataset consisting of 12,600 fake news articles and an equal number of authentic news articles. For the new dataset, they obtained an accuracy of 92% using both n-gram and LSVM. In their research, Ahmed, Traore, and Saad (2017) placed a primary emphasis on identifying opinion spam and fake news n-gram characteristics using various feature extraction techniques. The performance of the n-gram features was good on both real-world data and pseudo data, and it was even better when applied to the fake news data.

Table 2.5: Their approach to detecting opinion spam and fake news differs from previous studies.

Dateset	Classifier	Features	Performance Metrics	Score	Reference
Reviews Amazon website	Logistic regression	Review and reviewer features	AUC	78%	Jindal and Liu (2008)
Dataset 1 (Ott et al. (2011) reviews dataset)	SVM	LIWC+Bigrams	Accuracy	89%	Ott Choi Cardie & Hancock (2011)
Dataset 1 (Ott et al. (2011) reviews dataset)	SVM	Stylometric features	F-measure	84%	Ott Choi Cardie & Hancock (2011)
Dataset 1 (Ott et al. (2017) reviews dataset)	LSVM	Bigram	Accuracy	90%	Ahmed, Traore, and Saad (2017)
Dataset 2 (Ahmed, Traore, and Saad (2017) news dataset)	LSVM	Unigram	Accuracy	92%	Ahmed, Traore, and Saad (2017)
Buzzfeed news and random new articles (Horne and Adali's news dataset)	SVM	Text-based features	Accuracy	71%	Mukherjee Venkataraman , Liu and Glance (2013)
Buzzfeed news and random new articles (Horne and Adali's news dataset)	LSVM	Unigram	Accuracy	87%	Ahmed, Traore, and Saad (2017)

Hosseinimotlagh and Papalexakis (2016) aimed to categorize fake news by using clustering techniques. Their proposal involved utilizing tensor modelling to detect fake news. This approach allowed them to capture the interrelation of both the articles and terms, as well as the spatial and contextual relationships between terms, ultimately revealing the full potential of the content. In addition, they introduced an ensemble approach that effectively merges and integrates outcomes from various tensor decompositions to produce distinct and precise clusters of articles that fall into different groups of fake news. By testing their proposed approach on labelled real-world data, the researchers showed that the algorithm could accurately classify all types of fake news present in the corpus. On average, the method achieved a homogeneity of approximately ~80% for each category. By modelling the context of words and their spatial vicinity within a document, the researchers were able to achieve high accuracy in discerning coherent sets of articles that pertain to various types of false news.

In order to achieve their objective, the researchers employed a third-order tensor representation to capture the interplay between articles and terms, as well as the spatial and contextual relationships among terms. Their findings emphasized the significance of leveraging both aspects of the corpus, particularly focusing on the spatial relationships between words, to effectively identify coherent clusters of articles associated with different types of false news. To facilitate the grouping of articles, a clustering algorithm was employed. This algorithm classified the articles into distinct clusters by leveraging their membership within a collection of diverse tensor decompositions. More specifically, the algorithm aimed to identify a subset of news articles that frequently formed clusters across various configurations of the first-tier tensor decomposition.

The arrangement of news articles belonging to the same latent factor is illustrated in Figure 2.6, where they are positioned contiguously within the vector.

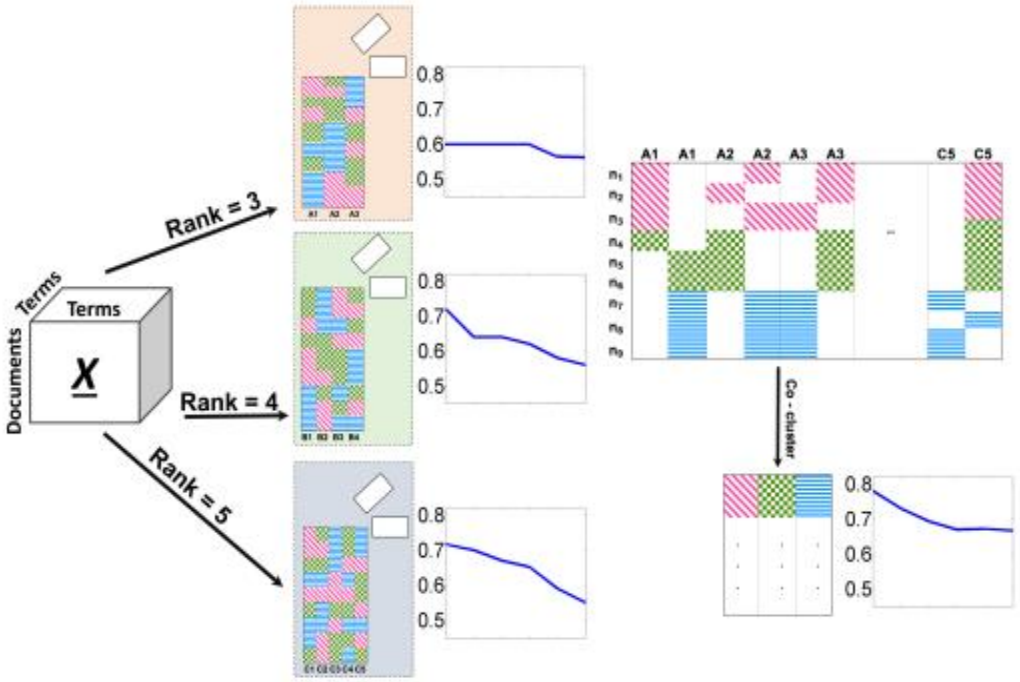

Figure 2.6: The proposed algorithm's summary outlines the formation of the S matrix through an ensemble of tensor decompositions and the extraction of top-notch clusters for counterfeit news articles. Source: Hosseinimotlagh and Papalexakis (2016).

Yang et al. (2019) employed a Generative Approach in their pursuit of detecting and identifying instances of bogus news on social media platforms. The central concept involves using users' interactions with news tweets on social media as auxiliary information to extract their viewpoints on the news. These viewpoints are then aggregated in a carefully planned unsupervised approach to generate estimation outcomes. According to Pang and Lee (2008), users' responses to news tweets make it possible to uncover their viewpoints on the news. Based on that intuition, Yang et al (2019) strived to leverage users' opinions on news, as inferred from their engagement behaviours on social media, to discern the authenticity of the news.

Figure 2.7 provides an illustrative depiction of the hierarchical user engagement model in the realm of social media, offering a comprehensive overview of its structure and components. In particular, numerous news tweets related to each news article in the news corpus can be identified and gathered from various social media platforms.

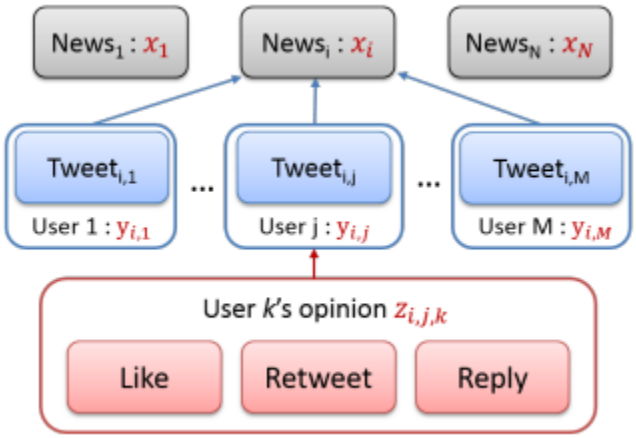

Figure 2.7: Hierarchical User Engagement Model. Source: Yang et al (2019).

The researchers inferred the authenticity of a news article by analysing the users' engagement behaviours and extracting their opinions on the news. For each given news *i,* they used a latent random variable $x_i \in \{0,1\}$ to convey the truth, i.e., false news (x_i = 0) or genuine news (x_i = 1). They let $y_{i,j} \in \{0,1\}$ to represent the user's opinion on the news and the opinion of the unverified user as $z_{i,j,k} \in \{0,1\}$.

Given the definitions of x_i , $y_{i,j}$ and $z_{i,j,k}$ they then presented their unsupervised false news detection framework (UFD). The model's probabilistic graphical structure is illustrated in Figure 2.8, with each node in the graph representing a random variable. The parameter θ represents a parameter from Bernoulli distribution and β and α represents hyperparameters from beta distribution. For each authenticated user, their credibility in the identification of fake news is modelled. More precisely, Ø denotes their sensitivity, signifying the true or positive rate in the context of fake news identification. The symbol Ψ denotes the probability that an unverified user perceives the news as true, given the contextual considerations of both the truth estimation of the news and the opinion expressed by a verified user.

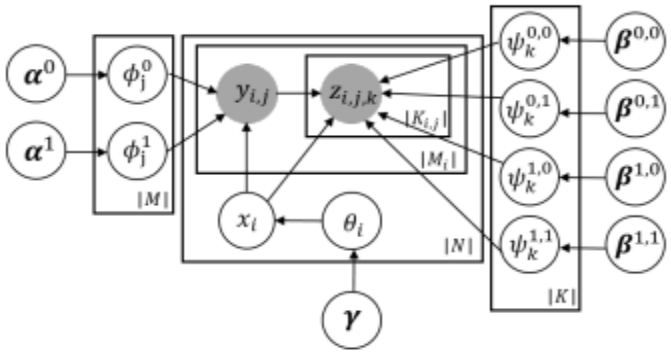

Figure 2.8: The Probabilistic Graphical Model. Source: Yang et al (2019).

The researchers used two publicly available datasets to test their model, LIAR (Wang 2017) and BuzzFeed News. Table 2.6 shows the counts of the test datasets.

Table 2.6: The statistics of datasets. Source: Yang et al (2019).

Datesets	LIAR	BuzzFeed
# News	332	144
# True news	182	67
# Fake news	150	77
# Tweets	2589	1007
# Verified users	550	243
# Unverified users	3767	988
# Engagements	19769	7978
# Likes	5713	1277
# Retweets	10434	2365
# Replies	3622	4336

Table 2.7 shows the experiment results on the BuzzFeed and LIAR datasets, respectively. Different measurements (Recall Precision and F1-score) were used to measure each news class.

UFD demonstrates the highest level of performance on the LIAR dataset, exceeding the accuracy of the second-ranked algorithm by 18.4% on the datasets. On the BuzzFeed dataset, except for recall on the fake news category, UFD attained the top-level performance.

Table 2.7: A comparison of performance was conducted on the LIAR and BuzzFeed datasets. Source: Yang et al (2019).

Methods	Accuracy	TRUE			Fake			Dataset
		Precision	Recall	F1-score	Precision	Recall	F1-score	
Majority Voting	0.586	0.624	0.628	0.626	0.539	0.534	0.537	
TruthFinder	0.634	0.65	0.679	0.664	0.615	0.583	0.599	
LTM	0.641	0.654	0.691	0.672	0.624	0.583	0.603	LIAR
CRH	0.639	0.653	0.687	0.669	0.621	0.583	0.601	
UFD	**0.759**	**0.766**	**0.783**	**0.774**	**0.75**	**0.732**	**0.741**	
Majority Voting	0.556	0.532	0.373	0.439	0.567	0.714	0.632	
TruthFinder	0.554	0.523	0.359	0.426	0.568	0.72	0.635	
LTM	0.465	0.443	0.582	0.503	0.5	0.364	0.421	BuzzFeed
CRH	0.562	0.542	0.388	0.452	0.573	0.714	0.636	
UFD	**0.679**	**0.667**	**0.714**	**0.69**	**0.692**	**0.43**	**0.668**	

Dusmanu, Cabrio, and Villata (2017) used a different approach to identify facts from opinions. The researchers used a method called Argument Mining. Argument Mining refers to the process of automatically extracting arguments and their relations from various textual corpora. This involved extracting natural language arguments and representing them in a structured form that can be used by computational models and reasoning engines (Peldszus and Stede, 2013; Lippi and Torroni, 2016). Since Twitter data has short texts and non-standard spelling with specific conventions (e.g., hashtags, emoticons), it represents a challenge for argument mining approaches (Snajder, 2016). The distinct nature of social media data necessitates the definition of new tasks in the field of argument mining:

- arguments extraction
- relations prediction

Argument extraction involves getting an argument from within the input natural language texts and then detecting its boundaries. The purpose of relation prediction is to anticipate the relations between the arguments recognized during argument extraction. The researchers used a supervised model to separate argument tweets from non-argumentative tweets. They did this while using Grexit and Brexit news datasets. A dataset consisting of the vote to decide whether Greece should remain in or leave the European Union (EU) was used as the first dataset (the thread #Grexit with 987 tweets), and a dataset consisting of the referendum for or against Britain leaving the EU was used as the second dataset (the thread #Brexit with 900 tweets). Table 2.8 displays the outcomes of the initial task (argument detection) on the two datasets.

Table 2.8: Dataset for task 1: argument detection. Source: Dusmanu, Cabrio and Villata (2017).

dataset	# factual arg.	# opinion	total
Brexit	713	187	900
Grexit	746	241	987
Total	1459	428	1887

The researchers used Logistic Regression (LR) and Random Forest (RF) classification algorithms, table 2.9 shows the obtained results.

Table 2.9: The outcome achieved on the argument detection task's test set is as follows (L=lexical features). Source: Dusmanu, Cabrio and Villata (2017).

Approach	Precision	Recall	F1
RF + L	0.76	0.69	0.71
LR + L	0.76	0.71	0.75
LR + all Features	0.81	0.779	0.78

Table 2.10 shows results obtained from performing the second task of argument mining. The purpose of this undertaking is to classify argumentative tweets as either information-based or opinion-based, as described by Park et al. (2015).

Table 2.10: The dataset intended for Task 2 involves distinguishing between factual arguments and opinions classification. Source: Dusmanu, Cabrio and Villata (2017).

dataset	# factual arg.	# opinion	total
Brexit	138	575	713
Grexit	230	516	746
Total	368	1091	1459

Table 2.11 displays the precision, recall and F1 obtained.

Table 2.11: The outcomes achieved on the test set for the task of classifying factual versus opinion arguments (L=lexical features). Source: Dusmanu, Cabrio and Villata (2017).

Approach	Precision	Recall	F1
RF + L	0.75	0.68	0.71
LR + L	0.75	0.75	0.75
LR + all Features	0.81	0.79	0.8

Ferrara and Montanelli (2017) in their research proposed using attraction to topics (A2T), a technique for identifying argumentative discourse units at the level of individual sentences is presented, utilizing unsupervised methods based on topic modelling.
The attraction to topics was used for two reasons:

- Attraction to topics (A2T) not only identifies sentences with argument components but also distinguishes them from non-argumentative sentences that lack argument components.
- Attraction to topics (A2T) categorizes the discovered argumentative sentences into three groups major claims, claims, and premises.

The concept of topic attraction suggests that an argumentative unit is a sentence that is particularly concentrated on a specific topic. In other words, it refers to a sentence with a high assignment value for a specific topic and a low assignment value for other topics. This implies that the concept of attraction is employed to identify sentences that are highly focused on specific topics, in order to recognize argumentative units.

Figure 2.9 shows the schema of the attraction to topics approach. As shown it starts with corpus of texts C ={C_1……..C_n}.The final step of the attraction to topics approach is to derive a set of argumentative units U = {<s_1, c, l>, . . . ,< s_n, c, l>}, where S_i represents a sentence containing an argumentative unit, *C* denotes the text that includes *S*, and l indicates the expressed argumentative role of the unit.

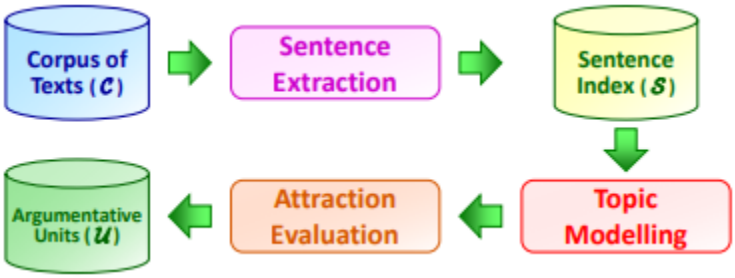

Figure 2.9: Schema of the attraction to topics (A2T) approach. Source: Ferrara and Montanelli (2017).

Figure 2.10 shows the precision, recall, and F1-measure for attraction to topics (A2T) and for the baseline. The results of applying sentence labelling based on ϕ and ρ (the components of attraction) independently. The parameter ϕ represents the distribution of words over topics, while ρ is a measure of how much S_i is focused on a topic. From Figure 2.10, we can also see that attraction to topics (A2T) achieves a significantly higher F1-measure than the baseline, particularly for the C1 corpus. Both the attraction components, ϕ and ρ, exhibit good performance, with the topic component ϕ showing slightly better precision and recall compared to the position information ρ. Similar results are observed for the C2 corpus, with A2T outperforming the baseline, albeit with a slightly smaller precision advantage.

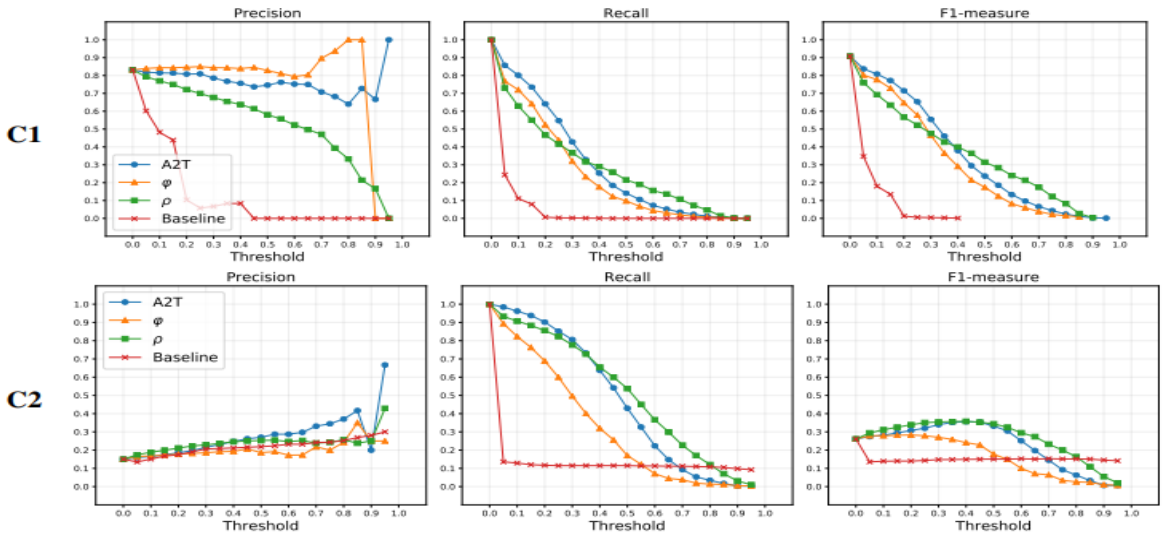

Figure 2.10: Precision, recall, and F1-measure were computed at varying thresholds. Source: Ferrara and Montanelli (2017).

In summary, the evaluation results on two different corpora are shown above. It is evident that A2T outperforms the baseline, particularly for the C1 corpus. The blue line is above the red line throughout the different thresholds. This corpus is distinctive in the sense that argumentative units are often situated in the essay's introduction or conclusion. Ferrara and Montanelli (2017) showed in their paper that A2T outperformed the baseline for argumentative sentence detection on both corpora.

In an alternative method expounded by Asha and Meenakowshalya (2021), the differentiation between counterfeit and authentic news was realized through the application of an Unsupervised learning algorithm, specifically the One-Class SVM (Support Vector Machine). The One-Class SVM represents a variant of the Support Vector Machine paradigm. Functioning as a one-class classifier, the algorithm discerns a hyperplane within the news dataset, positioned in proximity to the origin, with the objective of minimizing the spatial separation from the dataset's data points (Asha and Meenakowshalya, 2021). Within their proposed approach, the Unsupervised algorithm yielded an accuracy rate of 54.67%. Within this chapter, the concept of fake news was formally delineated, followed by an examination of several studies that employed both supervised and unsupervised approaches to discern the veracity of an article.

The research presented reveals that the prevailing methodologies for detecting false news predominantly lean towards supervised methods, which necessitate significant investments of time and resources in order to construct meticulously annotated datasets. The insights and concepts elucidated throughout this chapter will serve as valuable considerations in the subsequent chapters of this research, where original experiments will be conducted. The next chapter will describe the research design methods.

CHAPTER 3: RESEARCH METHODOLOGY

INTRODUCTION

This section of the study will cover the research design, data collection methods, data measurement, and analysis. According to Coldwell and Herbst (2004), research design encompasses the methodological approach employed in conducting research, which can be exploratory, descriptive, or causal depending on the stage of the research topic. The research design will be determined by the stage to which the research topic has progressed (Sekaran and Bougie, 2013).

3.1 DATA COLLECTION

3.1.1 DESCRIPTION

Tweets, also known as "Twitter data “, were used as the primary data source for this study. Primary data refers to first-hand information (Sekaran and Bougie, 2013). As of March 2022, a simple search for "Twitter data" on Google Scholar yielded over 74,000 results, indicating the platform's attractiveness for academic research. According to BusinessOfApps (2022), Twitter has approximately 300 million users, with over 75% of them active on a daily basis. Twitter provides a wealth of data for research, including up to 500 million tweets per day and up to 90 variables for each tweet (Umit, 2022). Most Twitter data is publicly available, although there are relatively few private profiles.

3.1.2 ETHICAL CONSIDERATIONS

Voluntary participation: Each and every participant acted voluntarily, and there was no use of force or coercion to encourage them to share a tweet about COVID-19. Informed consent: All participants (users) of Twitter sign the terms and conditions with Twitter which consents to the use of any data they post “By providing, uploading, or presenting Content on or through the Services, you confer upon us a broad license with global reach, non-exclusive rights, and royalty-free terms, including the authority to grant sublicenses.

This license enables us to utilize, duplicate, reproduce, process, adapt, modify, publish, transmit, display, and distribute the content through any existing or future media or distribution methods, encompassing activities such as curation, transformation, and translation. It is important to note that this license allows us to share your Content with the global audience and authorize others to do the same. By agreeing to these terms, you acknowledge that this license empowers Twitter to provide, promote, and enhance the Services, as well as to facilitate the availability of Content submitted via the Services to other companies, organizations, or individuals. These entities may engage in activities such as syndication, broadcast, distribution, Retweeting, promotion, or publication of the Content on various media and services. However, any such utilization is subject to the terms and conditions set forth by Twitter. It is important to understand that these additional uses of your Content by Twitter or other entities do not entail compensation to you. Your usage of the Services is deemed sufficient compensation for the Content provided and the rights granted herein.", (https://twitter.com/en/tos, 2022).

Confidentiality and respect: Privacy was ensured for every participant (user) and their tweets, as well as respect for their autonomy.

Data safety: Data was safely stored during and after the research project to ensure its security.

Data integrity: The reporting of data collected during the study was accurate and did not involve any misrepresentation or distortion.

Subject safety: The participants (users) were not exposed to any mental or physical harm, and users shared their tweets with Twitter willingly.

3.1.3 SAMPLING

According to Sekaran and Bougie (2013), since it is not feasible to gather data from every individual in a population, a sample - which is a smaller subset of the population - must be used instead (or all data available from Twitter). Thus, sampling the population becomes important. Twitter has restrictions on data access, mainly two restrictions:

- The amount of data that can be downloaded.
- At what rate, how often, and how far back in time can the data be accessed.

These restrictions vary across API types, such as Standard v1.1, Premium v1.1 (or Enterprise: Gnip 2.0), and Academic Research access. These restrictions also differ among API types and across various operations, e.g., the restrictions may vary based on whether one is collecting real-time tweets or historical tweets, as well as whether one is collecting historical tweets from a specific user or from any user. We chose the Standard v1 API type for the study. The researcher was able to download 878,721 COVID-19 tweets, which is just a sample of all the COVID-19 tweets available. All COVID-19 tweets written in the English language were considered; this was not limited to any region. The sampling method used to extract these tweets was convenience sampling. Battaglia (2013) provided the definition of convenience sampling as a form of non-probability sampling that involves selecting individuals or items as sources of data for research solely based on their availability.

3.2 EXPLORATORY DATA ANALYSIS

3.2.1 SENTIMENT ANALYSIS

Often it is not solely significant to comprehend the content of what users are expressing, but rather the way they are expressing it is also crucial. The objective of "Sentiment Analysis" is to automatically link a text excerpt with a "sentiment score," which indicates a positive or negative emotional score. Summarizing sentiment can provide insight into the overall response of individuals towards a particular company, product, or topic. The practice of employing Natural Language Processing (NLP) to detect and extract subjective information from a text is known as sentiment analysis. The information contained within a text can pertain to various emotions, including negativity, positivity, fear, joy, and so on. This becomes particularly valuable in gaining an overall “feeling” of the topics people are discussing on Twitter, such as the COVID-19 pandemic. The sentiment analysis algorithm employed in this analysis is a Naïve Bayes Classifier. As implemented in the package (tidytext), this algorithm categorizes a tweet as positive or negative by matching each word in the tweet with the labelled words present in the lexicon (Kumar, Morstatter and Liu, 2014). If the vocabulary used in a tweet primarily aligns with that commonly found in positive tweets, the tweet is classified as positive.

Conversely, if the vocabulary is predominantly associated with negative tweets, the tweet is classified as negative.

3.2.2 WORD CLOUDS ANALYSIS

Very often, word clouds are used to analyse twitter data or a corpus of text. Word clouds are a useful summarization technique that visually emphasize significant words in the text, with the frequency of occurrence used as an indicator of their importance. Word cloud is beneficial in that it aids in the identification of the connection between search terms and the diverse words contained in the data frame. Word clouds are created by taking a set of text data and processing it to remove common words like "the" "and" and "of" as well as punctuation and other nonessential elements. Subsequently, the remaining words are organized based on their frequency, wherein the words that occur most frequently are visually emphasized through a larger font size, while words with lower frequencies are visually represented in a smaller font size. Word clouds are excellent tools for visualization that present textual data in an uncomplicated and transparent format. Word clouds serve as efficient and concise visual representations utilized to encapsulate the principal themes or topics within a given text or dataset. In the next chapter, we will perform a cluster analysis to classify COVID-19 news-related tweets as fake or real. It will be interesting to see which words are associated with fake or real COVID-19 tweets.

3.3 NATURAL LANGUAGE PROCESSING (NLP) MODELLING

According to IBM (2020), Natural Language Processing (NLP) pertains to the field of computer science and machine learning that focuses on endowing computers with the capability to comprehend spoken language and written text in a manner akin to human cognitive processes. Briefly, NLP is the ability of computers to understand human language. The primary challenge in language processing is that machine learning algorithms cannot process raw text directly, so word embedding solves this problem. The word embedding techniques are used to represent words mathematically (numerically).

According to Akdogan (2021) words need to be made meaningful for machine learning or deep learning algorithms. Once words are converted into their numerical representations, after learning these representations, machines can efficiently process textual information. There exist several common approaches to feature extraction, including Word2Vec, Term Frequency Inverse Document Frequency (TF-IDF) and Bidirectional Encoder Representations from Transformers (BERT) allow words to be expressed mathematically as word embedding techniques (Akdogan, 2021). Words are represented as numerical vectors with real values using word embeddings (Agarwal,2022). To achieve this, it breaks down every word in a sequence or sentence into tokens, then transforms them into a vector space. Tokenization plays a crucial role in handling text data in natural language processing (NLP), it involves dividing a block of text into smaller units, which are referred to as tokens, (Dahouda and Joe, 2021).

3.3.1 WORD EMBEDDING ALGORITHMS

3.3.1.1 TF-IDF

TF-IDF stands for Term Frequency and IDF stands for Inverse Document Frequency. TF-IDF is a technique that combines two separate terms, namely Term Frequency and Inverse Document Frequency. The Term Frequency measures the frequency of occurrence of a term in each document, while the Inverse Document Frequency reflects the number of times a specific word appears in a collection of documents. High-frequency words that appear in all documents are not considered important, hence IDF is used to measure the significance of a word in all documents. IDF is the inverse of the Document Frequency, and it helps to assign more weight to less commonly occurring words. According to Akdogan (2021), TF-IDF is a statistical measure utilized to assess the mathematical significance of words within documents. The process of vectorization is akin to One Hot Encoding, and the TF-IDF value is obtained by multiplying the TF (Term Frequency) and IDF (Inverse Document Frequency) values (Akdogan, 2021).

The mathematical pseudo code for TF-IDF is:

TF-IDF = TF * IDF

where:

TF = (Number of times term appears in a document) / (Total number of terms in the document)

IDF = log(Total number of documents / Number of documents with term in it)

Therefore, the final TF-IDF score for a term in a document is the product of the term's TF and IDF scores.

3.3.1.2 WORD2VEC

Word2vec (word to vector), is a frequently used word embedding technique. According to Karani (2020) Word2vec is a two-layered neural network that produces a model of relative word embedding, and it is commonly used to convert words into vector form. To create vectors, one must identify the words that appear more frequently in conjunction with the target word (Akdogan, 2022).

3.3.1.3 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT was introduced by Google in 2018 and belongs to a class of NLP-based language algorithms known as transformers (Agarwal, 2022). BERT is a transformer-based model that is pre-trained on a large scale and bidirectional in nature. It is available in two models, the one model is BERT-Base and has 110 million parameters, and the other is BERT-Large and has 340 million parameters (Agarwal, 2022).

According to Pogiatzis (2019) the two models are as follows:

- The BERT-Base: Number of transformer blocks (L): 12, Hidden layer size (H): 768 and Attention heads(A): 12
- The BERT-Large: Number of transformer blocks (L): 24, Hidden layer size (H): 1024 and Attention heads(A): 16

According to Rajapaksha (2020), it was posited that BERT exhibits a comparative advantage over models such as Word2vec due to its capacity to generate word representations that are dynamically influenced by the neighbouring words. In contrast, Word2vec relies on fixed word representations, independent of the contextual environment in which the word occurs. In this study all three distinct word embedding techniques were incorporated during the experiments.

3.3.2 UNSUPERVISED LEARNING TECHNIQUES

Since the beginning of the Artificial Intelligence (AI) implementation, many techniques have been used, and many others are emerging until this day.
This subchapter discussed two different unsupervised clustering techniques and two different dimensionality reduction methods were discussed.
For clustering:

- K-means clustering.
- Mini-Batch K-Means clustering.

For dimensionality reduction:

- t-distributed stochastic neighbourhood embedding (t-SNE).
- Truncated singular value decomposition (SVD).

3.4 CLUSTERING

3.4.1 K-MEANS CLUSTERING

As previously indicated, K-means clustering is an unsupervised methodology employed in scenarios where the data lacks labels. The primary objective of this technique is to identify clusters or groups within the unlabelled data, with the number of clusters denoted by K. In summary, K-means clustering divides unlabelled data into K clusters in a way that each cluster has at least one object (data point). The nomenclature "k-means" was originally introduced by James Macueen in 1967, as documented by Baria and Yadav (2014). However, the conceptual origins trace back to Hugo Steinhaus in 1957.

The foundational algorithm, initially conceived by Stuart Lloyd in 1957 for pulse code modulation, remained unpublished until 1982. In 1965, E.W. Forgy independently introduced a method essentially akin to Lloyd's, sometimes denoted as Lloyd-Forgy. A more efficacious iteration was subsequently presented and documented in FORTRAN by Hartigan and Wong in the years 1975 to 1979. Forgy (1965) published a method similar to the one proposed by Lloyd, which is why the method is sometimes referred to as Lloyd-Forgy (Forgy, 1965). According to Qi et al. (2017), K-means is a well-established and effective method for clustering data. The algorithm operates on a dataset D= {p_i |*i* = 1, ..., n}, p_i in d-dimensional space R^d. K-means selects K seeds, which serve as the initial centroids for K clusters. The K-means algorithm then assigns each point to a cluster by minimizing the sum of squared errors (SSE). The function of K-means is demonstrated in Equation (1), where **|** $p_i - m_i$ | denotes the distance between point p_i and centre m_i of cluster C_j's, and this can be calculated using Equation (2).

The SSE formula is:

$$\text{SSE} = \sum_{j=1}^{k} \sum_{i=1}^{n} \delta_{ij} ||p_i - m_i||^2 \quad (1)$$

(δ_{ij} = 1 if $p_i \in C_i \in C_i$ and) otherwise)

$$\mu = \frac{\sum_{p_i \in C_j} p_i}{|C_j|} \quad (2)$$

The pseudo-code for K-means according to Rosenberg and Hirschberg (2007) is shown in table 3.1:

Table 3.1: K-means algorithm.

Algorithm 1: K-means

arbitrarily choose an initial k centres $C = \mu_1, \mu_2 \ldots\ldots\ldots\ldots\ldots, \mu_k$

repeat

for $i \in 1, \ldots\ldots\ldots\ldots, k$ **do**

set the clusters C_i to be the set of points in X that are closer to C_i than they are to $C_j \ \forall \ j \neq i$

end

for $i \in 1, \ldots\ldots\ldots\ldots, k$ **do**

set C_i *to be the center of mass of all points in* $C_i : \ c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

end

until *C* no longer changes;

Yuthika (2018) stated that there are pros and cons to using K-means clustering.

Pros:

1. K-means is simple, flexible, and efficient. Its simplicity makes it easy to explain the results compared to Neural Networks.
2. The flexibility of K-means technique makes it easy to adjust if needed. Its inherent flexibility allows for customization to address specific challenges, thereby endowing it with considerable potency. Whether confronted with structured data, embeddings, or any other data format, the incorporation of K-means is recommended for comprehensive and effective clustering solutions.
3. The efficiency of the K-means technique makes it well-suited for effectively segmenting datasets. The efficiency mentioned to the ability of the K-means clustering algorithm to perform its task in a resource-effective manner. In this context, efficiency may encompass several aspects, including computational speed, memory usage, and the algorithm's capability to handle large datasets.

Cons:

1. Determining the appropriate number of clusters for analysis necessitates a decision to be made beforehand. The selection of the number of clusters remains subjective and typically involves factors such as common sense, domain expertise, and related considerations.
2. The K-means technique entails the random selection of diverse initial cluster centroids, which can yield either advantageous or detrimental outcomes contingent upon the specific initial choices made.
3. The order of the unlabelled data has an impact on the final clusters.

3.4.2 MINI BATCH K-MEANS CLUSTERING

Mini-Batch K-means is recognized as a variant of the K-means algorithm, exhibiting a similar operational framework. Its distinctive feature lies in the utilization of small, random subsets, referred to as mini-batches, from the input data during each iteration of training (Sculley, 2010). Maharjan (2018) emphasizes that the Mini-Batch K-means algorithm capitalizes on these mini-batches to expedite computational processes while striving to optimize the same objective function as the original K-means algorithm.

The inclusion of mini-batches significantly reduces the computational burden involved in achieving convergence and finding an optimal solution (Maharjan, 2018).

Notably, Chavan et al. (2015) have demonstrated the enhanced performance of the Mini-Batch K-means clustering algorithm. They further elucidate the advantages of the Mini-Batch K-means Algorithm over the standard K-means Algorithm, as follows:

1. The utilization of mini batches in the Mini-Batch K-means Algorithm leads to a substantial reduction in computational time while still striving to optimize the same objective function. Baria and Yadav (2014) observe that Mini-Batch K-means comes at the expense of producing results that are typically inferior to those obtained from the standard algorithm (K-means).
2. The Mini-Batch K-means algorithm is a straightforward unsupervised learning technique employed for the purpose of addressing clustering conundrums.

According to Sculley (2010) the pseudo-code for Mini-Batch K-means can be expressed in the algorithm given in table 3.2.

Table 3.2: Mini-Batch K-means algorithm.

Algorithm 2: Mini-Batch K-means

Input: *k,* mini-batch size *b,* iterations *t,* dataset *X*

Initiate each $c \in C$ with an x picked randomly from X

$\mathcal{V} \leftarrow 0$

for $i = 1$ *to* t **do**

  M $\leftarrow$ *b* examples picked randomly from *X*

  for $x \in M$ do

    d[*x*] $\leftarrow f(C,x)$ // Cache the centre nearest to *x*

  end

  for $x \in M$ do

    $c \leftarrow$ d[*x*] // Cache the centre for this *x*

    $\mathcal{V}$[c] $\leftarrow \mathcal{V}$[c] +1 //Update per-centre counts

    $\boldsymbol{\eta} \leftarrow \frac{1}{\mathcal{V}[c]}$ //Update per-centre learning rate

    $c \leftarrow (1 - \boldsymbol{\eta})c + \boldsymbol{\eta}x$ take gradient step

  end

end

Mini-Batch K-means and K-means are clustering algorithms that will provide us with 'grouping' of the data in the form of clusters. The two clustering algorithms will not provide us with a visualization that can be used to interpret the clustering results. We will need to reduce the dimensions on the clustered data to two or three dimensions to allow us to be able to visualize these clusters.

3.5 DIMENSIONALITY REDUCTION

Dimensionality reduction serves as an exploratory data analysis (EDA) technique that facilitates the rapid visualization of data with high dimensions (Oskolkov, 2022). Additionally, it offers the prospect of unveiling latent systematic patterns inherent within a given dataset (Oskolkov, 2022). According to Vlachos (2011), dimensionality reduction refers to the technique of transforming an n-dimensional point into a lower p-dimensional space. This is a pivotal facet of the data mining process involves the imperative undertaking of dimensionality reduction. The escalation of dimensionalities markedly amplifies the intricacy and temporal demands inherent in clustering, diverging considerably from the task's execution in lower-dimensional spaces. This process is also beneficial for visualizing data, especially when objects are represented in two or three dimensions. Visual cluster analysis commonly utilizes dimensionality reduction methodologies to project high-dimensional data onto 2D scatterplots. Through this process, one can visually discern and identify cluster patterns within the data. The primary goal of dimensionality reduction is to uncover a mapping that can capture an efficient low-dimensional representation that is present within observable data of high dimensionality (Jia et al., 2022). Reducing the dimensionality enhances both the computational efficiency and precision of some data analysis. Kumar (2009) proposed a method to mathematically define dimension reduction, which involves seeking a lower-dimensional representation S= $(s_1, s_2, \dots \dots \dots s_k)^T$, where k < r, for a random vector X=$(x_1, x_2, \dots \dots \dots x_r)^T$. The objective is to maintain the essence of the original dataset as much as possible, based on a specific criterion.

According to Sivarajah (2020), dimensionality reduction is mainly used for:

- Data Compression
- Noise Reduction
- Data Classification
- Data Visualization

In this thesis we will use dimensionality reduction tools for our data visualization.

Two dimensionality reduction technique were employed to visualize the clustering results in this research namely t-distributed stochastic neighbourhood embedding (t-SNE) and Truncated singular value decomposition (SVD). t-distributed stochastic neighbourhood embedding (t-SNE) was created in 2008 by Van Der Maaten and Hinton. This technique, operating as a non-linear dimensionality reduction method, functions in an unsupervised manner, rendering it ideal for visualizing datasets with high dimensions (van der Maaten and Hinton, 2008). The fact that it is a nonlinear dimensionality reduction method means that this algorithm can separate data that cannot be separated by a straight line.

t-SNE aims to discover the inherent patterns within the data by considering the proximity of a sample's neighbours (Erdem, 2020). t-SNE turned out to be very computationally expensive, but it produced visualizations that were better than those produced by the other techniques (PCA, SVD etc) on most of the data sets it was tested on (Van Der Maaten and Hinton, 2008). Truncated SVD (Singular Value Decomposition) is a technique used for reducing the dimensionality of a high-dimensional matrix, resulting in a lower-dimensional representation of the original data. It works by finding the principal components of the matrix and only retaining the most important ones, hence "truncating" the rest. Truncated SVD implements a modified version of singular value decomposition (SVD), which computes the k most significant singular values, based on a user-defined parameter k (Vialas, 2018). The Truncated SVD is another name for the reduced SVD and is a computationally less complex form utilized for computing low-rank approximations (Manning, Raghavan and Schütze, 2008). Similar to PCA, Truncated SVD operates directly on the sample vectors, bypassing the need for covariance matrix computations (Meehan, 2020).

In summation, the utilization of dimensionality reduction techniques emerged as pivotal in facilitating the visualization of outcomes derived from K-means and Mini-batch K-means clustering. These techniques not only fostered the exploration, comprehension, and interpretation of high-dimensional data but also facilitated the analysis of cluster separation. Moreover, they furnished efficacious mechanisms for conveying the results of clustering endeavours.

The integration of dimensionality reduction as an integral visualization step contributed substantively to the comprehensive understanding and applicative efficacy of K-means and Mini-batch K-means clustering across diverse domains of research and data analysis.

CHAPTER 4: EXPERIMENTS

INTRODUCTION

In this chapter, we delve into the practical implementation of the study, focusing on the collection, processing, exploration, and utilization of the data to develop an unsupervised learning model for detecting COVID-19 fake news. The chapter is structured as follows, providing a clear flow and logic of the experiment:

- Data Collection and Summary:
  We begin by providing an overview of the data collection process employed for this study. This section outlines the sources from which the data was obtained, the criteria for data selection, and any pre-processing steps performed.
- Exploratory Data Analysis:
  Following the data collection, we conduct an exploratory data analysis to gain deeper insights into the dataset. This section encompasses two key analytical techniques: sentiment analysis and word cloud analysis.
- NLP Task and Word Embedding Methods:
  The subsequent section focuses on the primary task of natural language processing (NLP), specifically the detection of COVID-19 fake news. We present the methodology employed, which utilizes unsupervised learning techniques. The centre piece of this section is the utilization of various word embedding methods, which facilitate the transformation of textual data into numerical representations suitable for machine learning algorithms. We provide a detailed explanation of the selected word embedding techniques and their application to the dataset.
- Results:
  Lastly, we present the outcomes and findings obtained from the NLP task. This section encompasses the evaluation and analysis of the developed unsupervised learning model's performance in detecting COVID-19 fake news. The results are discussed, including any notable observations or trends identified during the analysis.

  Furthermore, the implications of the findings and their potential significance are discussed, contributing to the overall understanding of the study's objectives.

4.1 DATA: COVID-19 TWITTER DATA

In research, the fundamental element or the essence of research is data. Without data, research lacks substance.

4.1.1 DATA DESCRIPTION

In this section, we introduce the dataset that was used to cluster COVID-19 tweets as fake or real. Considering that clustering tasks are fully unsupervised, we concatenate all data collected. In this paper, all analyses were conducted using a sample of 500,000 COVID-19 tweets, which were selected from a total of 878,721 tweets downloaded. Table 4.1 indicates the minimum and the maximum dates the sample of 500,000 COVID-19 tweets were shared to the platform. The minimum date for our Twitter Data was 28-Feb-2022, while the maximum date is 05-Aug-2022.

Table 4.1: Minimum and Maximum of Date for Twitter.

Min & Max Date	Date & Time
Minimum Date	2022-02-28 10:44:54+00:00
Maximum Date	2022-08-05 08:21:50+00:00

The tweets were extracted using the R package twitteR, which was created by Michael W. Kearney from the University of Missouri. Twitter's Standard v1.1 APIs allow the extraction of a maximum of 5000 tweets at a time. Therefore, we spent several days extracting the different tweets to obtain a total of 500,000 COVID-19 related tweets.

4.1.2 EXPLORATORY DATA ANALYSIS (EDA)

Exploratory data analysis (EDA) can provide a useful starting point for analysing text data and generating meaningful insights. It can help to identify the main features and patterns within the text, which can then be used to guide further analysis and exploration.

Some common techniques for EDA in text data include:

- Sentiment analysis
- Word clouds analysis

SENTIMENT ANALYSIS:

To gain insight into how people are feeling about COVID-19, we used sentiment analysis. We used the get_nrc_sentiment function from the syuzhet library (Siswandi, 2017). The function uses Saif Mohammad's NRC Emotion lexicon (Jockers, 2020), which defines the sentiment groups shown in Figure 4.1. The score for each sentiment was then calculated, and we visualized our sentiment using the ggplot2 library (see Figure 4.1).

Figure 4.1 illustrates the sentiment of people behind the COVID-19 tweets in the sample. From the sentiment scores, it is apparent that most tweets related to the hashtag #covid19 are positive.

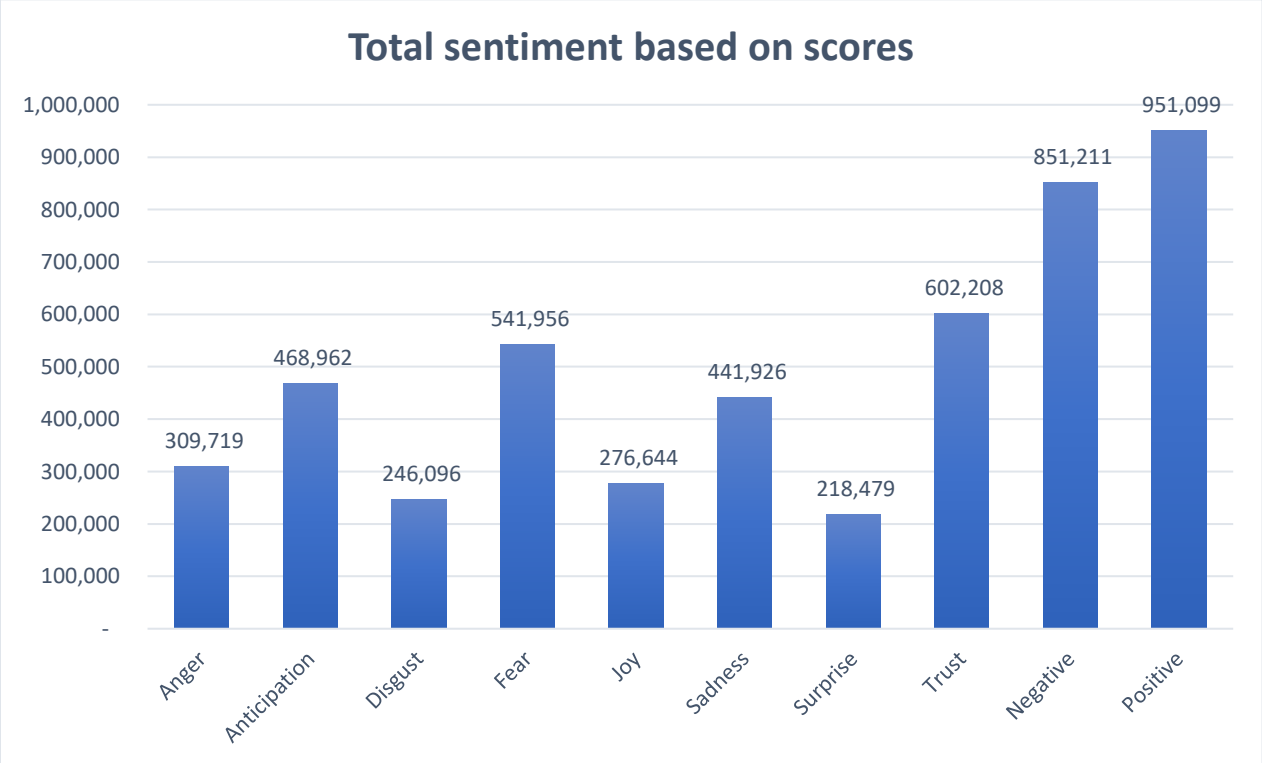

Figure 4.1: The Bar Plot displays the frequency of words in the text categorized by their emotional association.

Figure 4.2 illustrates the top 10 negative and positive words contributing to negative and positive sentiments. As expected, words like death, risk, sick, virus, etc., are all associated with negative sentiments. These words also make sense when one talks about COVID-19.

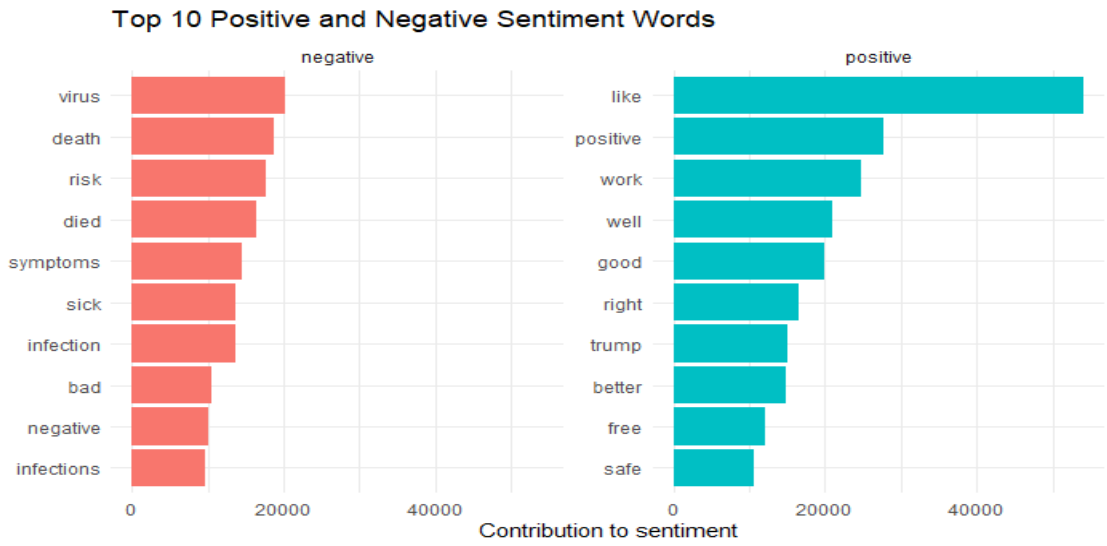

Figure 4.2: Top 10 Positive and Negative Sentiment Words.

As expected, words like safe, better, good, right and etc are all associated with positive sentiments. These words also make sense when talking about COVID-19 tweet.

WORD CLOUDS ANALYSIS:

Word clouds are a powerful visualization tool widely used to display data in various domains. Often, they are employed to create a visual representation of Twitter data or a corpus of text. Word clouds highlight the most significant words present in the text by showing their size according to their frequency. Thus, the frequency of each word becomes a crucial indicator of its importance, making word clouds an efficient summarization technique. Moreover, using a word cloud can help identify the relationship between search terms and the different words within the data frame. Figure 4.3 depicts the top 100 words, each appearing at least 50 times, in a word cloud. Words situated in the central region of the context exhibit a higher frequency, in contrast to those positioned on the periphery, which display a lower frequency. This representation allows for a quick overview of the most important terms in the text and their relative frequencies.

Words that are larger in size tend to have a higher frequency than those that are smaller in size. It must be expected that words like “death”, “cases” and “tests” occur more frequently when one talks about COVID-19.

Figure 4.3: Word cloud for COVID-19 Tweets (top 100 words).

In Figure 4.4, a "comparison.cloud" is depicted. This graphic displays a comparison of the most frequent words in tweets with positive and negative sentiment. As expected, words such as "death," "died," "virus," "risk," "burden," and "epidemic" are all associated with negative sentiment, whereas words like "free," "good," "best," "support," "right," "protect," and "safe" are all associated with positive sentiment. Later in this chapter, when we perform a cluster analysis, it will be interesting to investigate which words can be associated with fake or real COVID-19 news. This analysis may shed light on the ways in which Twitter users generate and disseminate information about the pandemic and help to differentiate between accurate and misleading information. By identifying the key themes and topics in COVID-19-related tweets, researchers and analysts can provide valuable insights into public perception, sentiment, and behaviour during a global crisis.

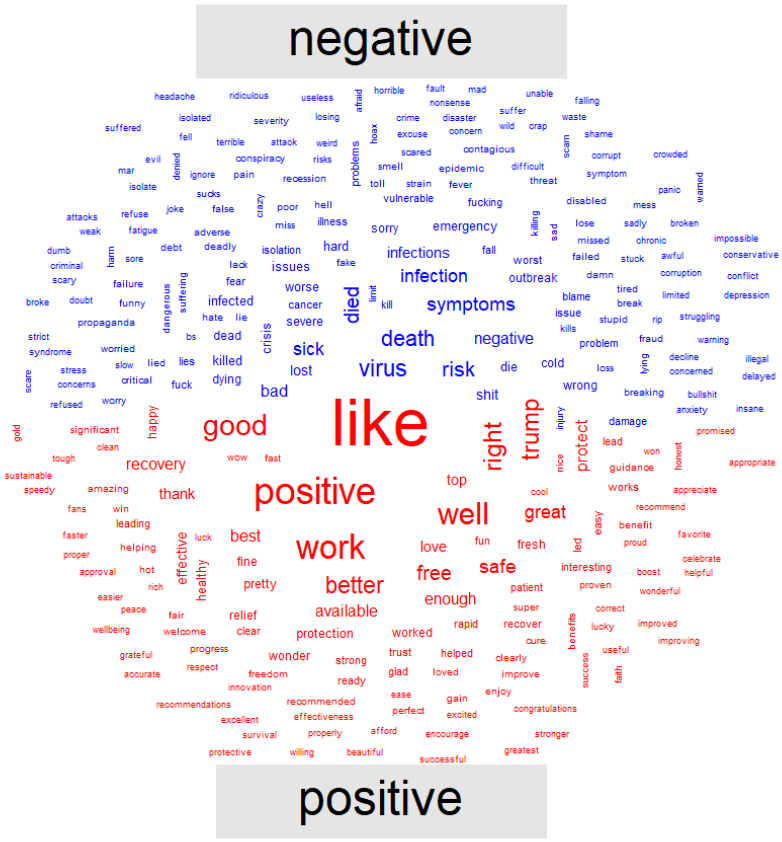

Figure 4.4: Word cloud for COVID-19 Tweets (top 200 words) with sentiment.

4.1.3 DATA PRE-POSSESSING

The Twitter data used for this research constitutes a semi-structured dataset. For the purpose of this thesis, only English tweets were included. However, this restriction may lead to an incomplete understanding of the situation. Moreover, due to practical constraints, it was not feasible to examine images, videos, or other associated media files, and only text data was utilized for this study. As shown in Table 4.2, the dataset includes a "text" field that contains tweets generated by users. However, these tweets may include noise, incomplete, and unreliable linguistic information. To analyse linguistic data from Twitter accurately, it is essential to eliminate this irregular data, so that the actual meaning and sentiments can be extracted from the dataset (Hemalatha et al., 2012). Therefore, data pre-processing was conducted to clean the dataset and ensure the accuracy and reliability of the analysis.

Table 4.2: Example Twitter data.

text
Major US study finds, #Vaccines for #Covid19 are not linked to deaths. https://t.co/GEuNQFges5
The sale of ultra-prime homes in #Dubai â€" those costing upwards of $10 million â€" made a post-#COVID19 comeback, totaling 93 for 2021 https://t.co/HziF75QxHz
Even mild Covid can shrink brain regions related to smell: Study https://t.co/9yKqRFVMFt #Brain #COvid #COVID19 #Ford #London
Travelling for work? If you are looking for Covid testing, please contact us on 033 022 02000, email us at info@southdownsprivatehealthcare.co.uk, or visit https://t.co/eHUchjCZqS. #Travel #Covid19 #CovidTest https://t.co/XwcSWBeZeB
"There are still so many causes worth sacrificing for.Â There is still so much history yet to be made." -@MichelleObama.
Itâ€™s Tuesday, March 8, 2022 & this is your daily reminder that it was @KamalaHarris who first told Americans NOT to get the #COVID19 vaccine=> @933KWTO @NewstalkSTL https://t.co/azBFrsPrYq
Globally #COVID19 vaccines must be approved by a @WHO listed regulator; @MHRAgovuk in the UK. See the listed regulators here: https://t.co/MHyLSX5qOg #VaccineCollaboration #VaccinesWork https://t.co/FLUkjKl56B
#COVID19 WW3 there are no better days ahead Jesus is coming soon
Meet Dr Trevor, our resident #Zwakala health professional who answers #COVID19 & vaccine-related questions in 3 local SA languages. #VaccinesSaveLives

The most generic view for pre-processing is shown by following Figure 4.5.

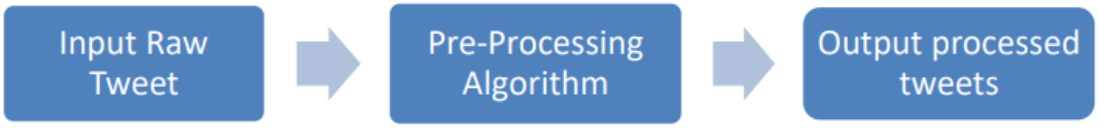

Figure 4.5: Overview of Data pre-processing.

During the data pre-processing phase, Fernández-Gavilanes et al. (2016) undertook several key steps, encompassing the normalization of linguistic data, noise reduction, and vocabulary simplification, all of which were essential for conducting sentiment analysis on tweets. According to Hemalatha et al. (2012), pre-processing informal text can enhance the performance of data mining tasks. In another study by Hemalatha et al. (2014), it was recommended to eliminate meaningless words to improve performance and obtain better results. This thesis presents a pre-processing algorithm for Twitter text data, based on the methodology proposed by Hemalatha et al. (2014). Hemalatha et al. (2012) delineates a methodologically sound approach in the pre-processing of Twitter data through a systematic sequence of steps. The study adeptly illustrates the intricacies of data preparation for machine learning techniques. The proposed algorithm aims to eliminate noise and irrelevant words in the dataset to improve the accuracy of the analysis results. By removing irrelevant words, the algorithm can identify the essential features and themes in the tweets, which may help in understanding public perception and behaviour during the COVID-19 pandemic.

The abstract idea for data pre-processing is show in Figure 4.6 below.

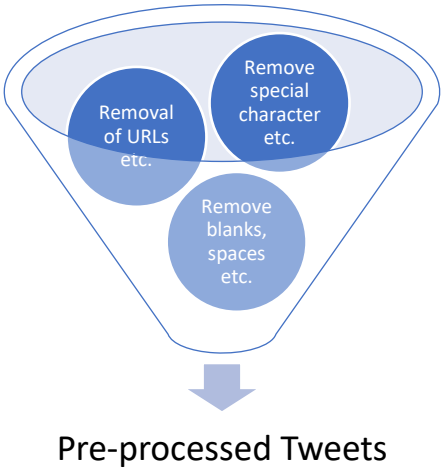

Figure 4.6: Structure for pre-processing user tweets on Twitter.

Table 4.3 shows the text data before pre-processing, symbols like "(: \ | [ ] ; : {} - + ( ) < > ? ! @ # % *,'URL')" are still contained in the data and after pre-processing when symbols are removed. Input Tweet refers to text data before pre-processing and Output Tweet refers to text data after pre-processing.

Table 4.3: Example of the dirty data before pre-processing and after pre-processing.

	text
Input Tweet	7 Day US Covid Deaths by County For MS 2022-03-04: Latest Covid Insights by Our Analytics Team using USAFacts #datavisualization #datascience #analytics #healthtech #data #covid19 #publichealth #covid #globalhealth #RStats
	POPUP news: update: DAILY DIM SUM: HKU research says total COVID-19 cases in HK reached its peak last Friday & will drop to double digits in mid-May
	Relevance and implications of health technology assessment for European public healthBy etelos1 from OC_gr aasuniversity UPHA_HTA UPHActs covering regulations
	Even mild Covid can shrink brain regions related to smell: Study #Brain #COvid #COVID19 #Ford #London
	Travelling for work? If you are looking for Covid testing
	"There are still so many causes worth sacrificing for.Â There is still so much history yet to be made." -ichelleObama.Since #COVID19 pandemic led to a significant rollback in #womensrights
	With news that ootsUK is selling lateral flow tests for Â£6 EACH
	One thing I've noticed ever since this RAT test thing came out(kit for testing your ownself-COVID) the amount of people at work just started to decrease in just less than a week. SMH. #COVID19
	7 Day US Covid Deaths by County For MS 2022-03-04: Latest Covid Insights by Our Analytics Team using USAFacts #datavisualization #datascience #analytics #healthtech #data #covid19 #publichealth #covid #globalhealth #RStats
	We have crossed SIX MILLION #COVID19 DEATHS
Output Tweet	7 Day US Covid Deaths by County For MS 2022-03-04 Latest Covid Insights by Our Analytics Team using USAFacts datavisualization datascience analytics healthtech data covid19 publichealth covid globalhealth RStats
	POPUP news update DAILY DIM SUM HKU research says total COVID-19 cases in HK reached its peak last Friday &amp will drop to double digits in mid-May
	Relevance and implications of health technology assessment for European public healthBy etelos1 from OCgr aasuniversity UPHAHTA UPHActs covering regulations
	Even mild Covid can shrink brain regions related to smell Study Brain COvid COVID19 Ford London
	Travelling for work? If you are looking for Covid testing
	There are still so many causes worth sacrificing forÂ There is still so much history yet to be made -ichelleObamaSince COVID19 pandemic led to a significant rollback in womensrights
	With news that ootsUK is selling lateral flow tests for Â£6 EACH
	One thing I've noticed ever since this RAT test thing came outkit for testing your ownself-COVID the amount of people at work just started to decrease in just less than a week SMH COVID19
	7 Day US Covid Deaths by County For MS 2022-03-04 Latest Covid Insights by Our Analytics Team using USAFacts datavisualization datascience analytics healthtech data covid19 publichealth covid globalhealth RStats
	We have crossed SIX MILLION COVID19 DEATHS

4.2 DATA: LABELLED DATA FOR TESTING MODELS

This section begins with an introduction of the dataset used to evaluate the accuracy of the six models discussed earlier.

Additionally, we provide a set of exploratory analyses performed on the test data, which are intended to provide a preliminary understanding of the dataset's characteristics and structure. The incorporation of distinct test data, focused on COVID-19 tweets, but independent from the training data, facilitates the evaluation of a trained model's capacity for generalization, the detection of potential issues such as overfitting or underfitting, the comparative analysis among diverse models, and ensures the essential aspects of transparency and reproducibility.

4.2.1 DATA DESCRIPTION

As an initiative of the 'First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation', which was conducted in conjunction with Constraint@AAAI (Association for Advancement of Artificial Intelligence) in 2021, a collaborative task was organized with the objective of identifying instances of COVID-19 misinformation in the English language. The shared task received 166 and 44 team submissions for the respective tasks, among which the BERT model and its variants emerged as the most successful models. The provided data consisted of three columns: id, tweet, and label. The label field denoted whether a tweet was fake or real, with all tweets pertaining to COVID-19 news. The test data is publicly available in the following link:https://github.com/archersama/3rd-solution-COVID19-Fake-News-Detection-in-English/blob/main/Constraint_English_Train.xlsx

4.2.2 EXPLORATORY DATA ANALYSIS (EDA)

To gain a deeper comprehension of the dataset, we conducted exploratory analyses on it, facilitating a more comprehensive grasp of the data at hand. Figure 4.7 illustrates the distribution split between fake and real news, 52% (3360) of these tweets were real tweets about COVID-19 news, while 48% (3060) were fake news about COVID-19.

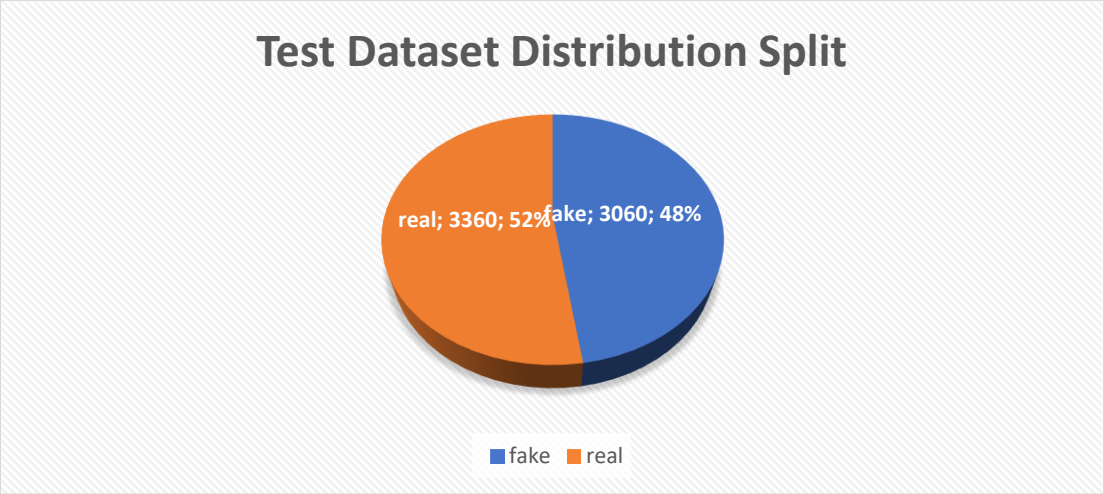

Figure 4.7: Test Dataset Distribution Split.

Based on the observations presented in Figure 4.8, it is apparent that within the test dataset, the words 'COVID', 'states', 'deaths', 'politically', and 'correct' exhibit the highest frequency.

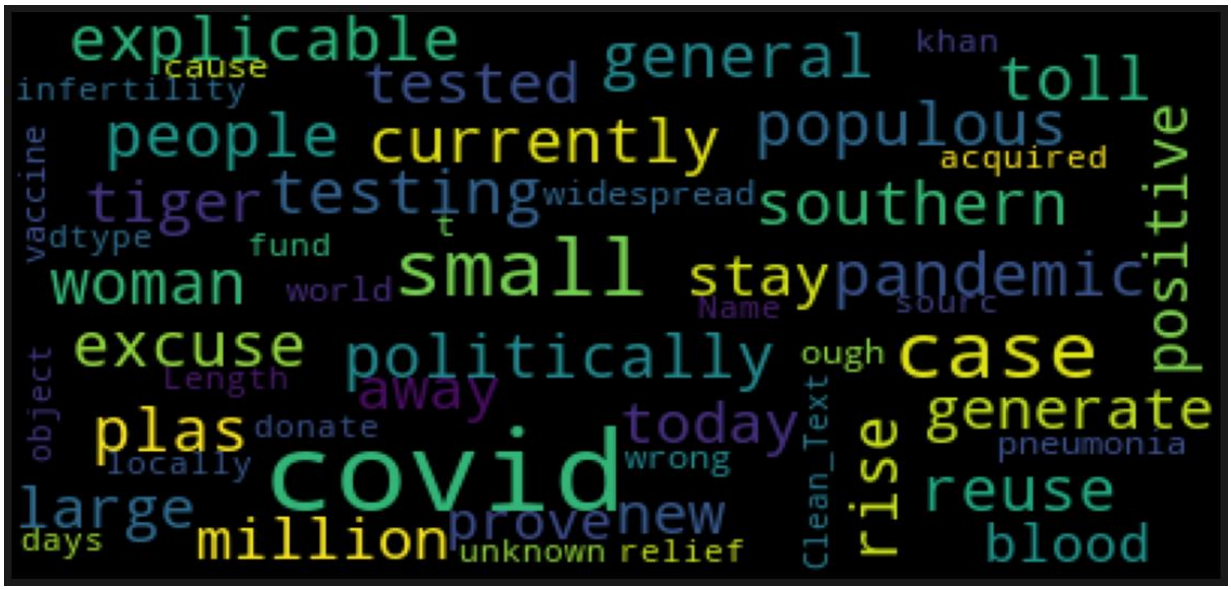

Figure 4.8: The word cloud diagram for test dataset.

Upon examining Figure 4.9, it becomes evident that within the test dataset characterized by the label "real," the words 'COVID', 'states', 'deaths', 'cases', and 'reported' exhibit the highest frequency.

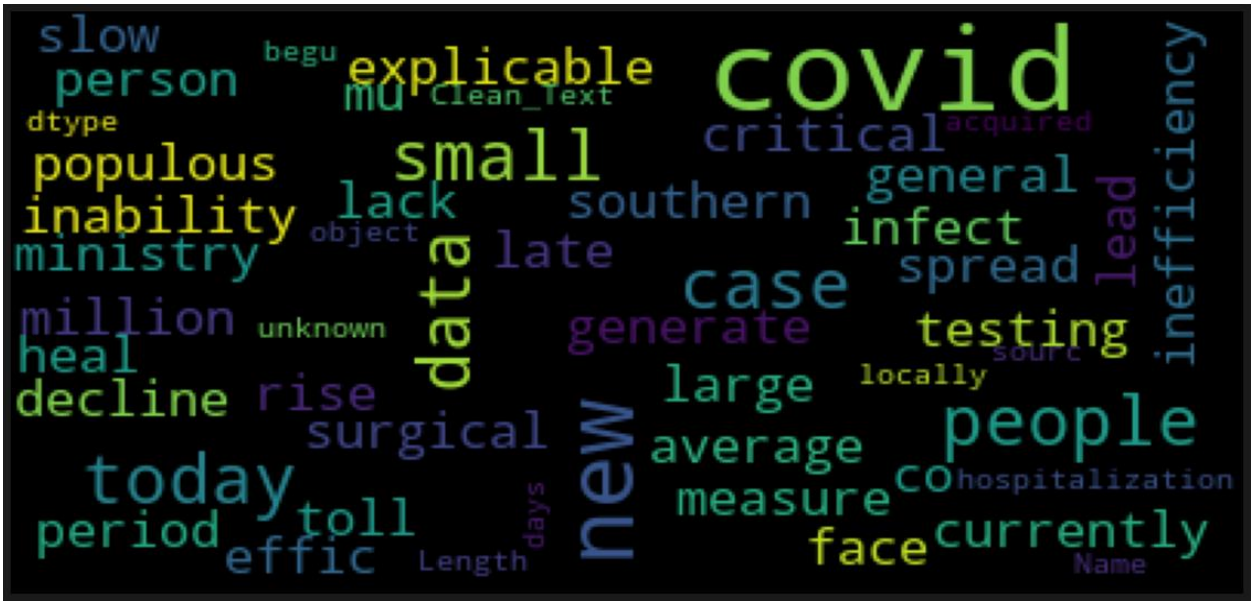

Figure 4.9: The word cloud diagram for test dataset with label= real.

As depicted in Figure 4.10, it is observable that within the test dataset denoted by the label "fake," the words 'COVID', 'Obama', 'Trump', 'calls', and 'politically' demonstrate the highest frequency. We can clearly see the differences between common word between fake and real tweets.

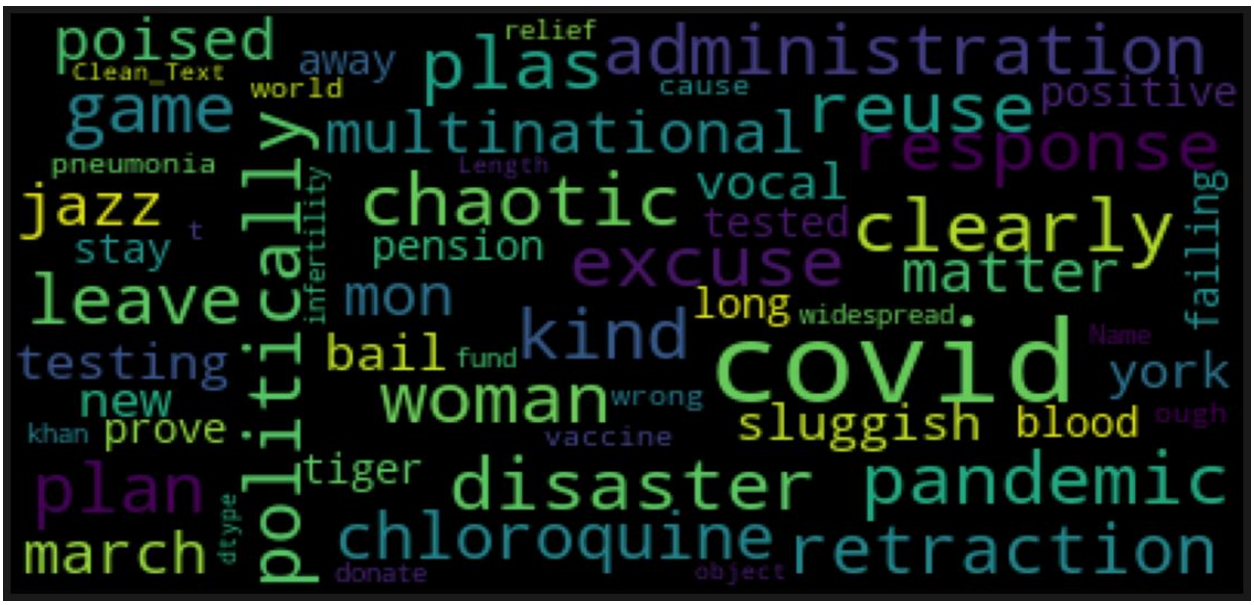

Figure 4.10: The word cloud diagram for test dataset with label= fake.

We conducted an evaluation of the predictions generated by our six models by means of basic visualizations, specifically employing confusion matrices.

4.3 NATURAL LANGUAGE PROCESSING (NLP) TASK: COVID-19 FAKE NEWS DETECTION

Both the Mini-Batch K-means and K-means algorithms were tested and compared with each other for a dataset with 500000 Tweets. We then compare the performance of the two clustering methods on real world data that has been labelled.

As explained in Chapter 3, we used 3 word embedding techniques to create features for use with the unsupervised clustering algorithms, Mini-Batch K-means and K-means. Six models for COVID-19 fabricated news detection were developed using the various combinations of the word-embedding techniques and clustering algorithms using the unlabelled data discussed in 4.1. The generated clusters were then analysed and profiled to see if they could indeed be said to represent sets of genuine and fake COVID-19 news. In addition, the unsupervised learning models were deployed on labelled data discussed in 4.2. This was done with the objective of assessing the generalization performance of the models when presented with novel data.

The following order will be followed for the discussion:

- Performance Analysis with Word2vec word embedding.
- Performance Analysis with BERT word embedding.
- Performance Analysis with TF-IDF word embedding.

4.3.1 PERFORMANCE ANALYSIS WITH WORD2VEC WORD EMBEDDING

Establishing the optimal number of clusters within a dataset constitutes a foundational concern in partitioning clustering methodologies, including k-means clustering. This necessitates the user to explicitly define the number of clusters, denoted as 'k,' to be generated. The Silhouette Score assumes significance in this context, serving as a metric for assessing the clustering quality and aiding in the determination of the optimal number of clusters. The average silhouette method calculates the mean silhouette of observations across various values of ‘k’.

The optimal number of clusters, denoted as ‘k’, is determined as the one that maximizes the average silhouette across a spectrum of potential values for k (Kaufman and Rousseeuw, 1990).

As articulated by Banerji (2023), the formula employed for computing the silhouette coefficient pertaining to a specific data point is expressed as follows: $s_i = \frac{a_i - b_i}{\max\{a_i, b_i\}}$

S(i) represents the silhouette coefficient of a given data point, wherein *a(i)* denotes the average distance between the said data point and all other data points within the cluster to which it is affiliated (Banerji, 2023). Concurrently, *b(i)* signifies the average distance from the data point to all clusters to which it does not belong (Banerji, 2023). Considerations to Bear in Mind During the Computation of the Silhouette Coefficient.

The silhouette coefficient assumes values within the range of [-1, 1]. An optimal score of 1 signifies the highest quality, indicating that the data point *i* is exceptionally cohesive within its affiliated cluster and distanced from other clusters. Conversely, the lowest attainable value is -1. Proximal values to 0 suggest the presence of overlapping clusters. Figure 4.11 depicts the determination of the optimal number of clusters through the Silhouette method, wherein the observation of the silhouette score reaching its maximum at K = 2 leads to the conclusion that the most suitable segmentation involves two clusters. Consequently, we shall proceed with the categorization into two clusters.

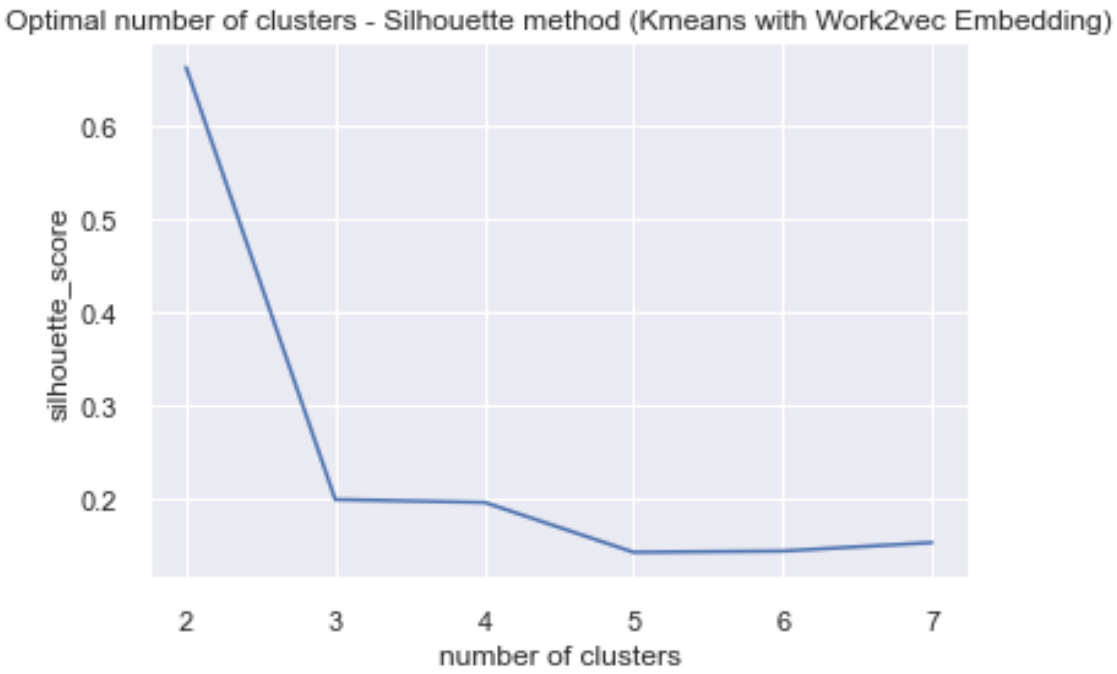

Figure 4.11: Optimal number of clusters - Silhouette method (Kmeans with Work2vec Embedding).

Figure 4.12 shows optimal number of clusters when using Silhouette method, observing the maximization of the silhouette score at K = 2, it is thereby determined that the optimal number of clusters is two, and as such, we shall proceed with the segmentation into two clusters for the Mini-Batch K-means clustering using a Word2vec embedding.

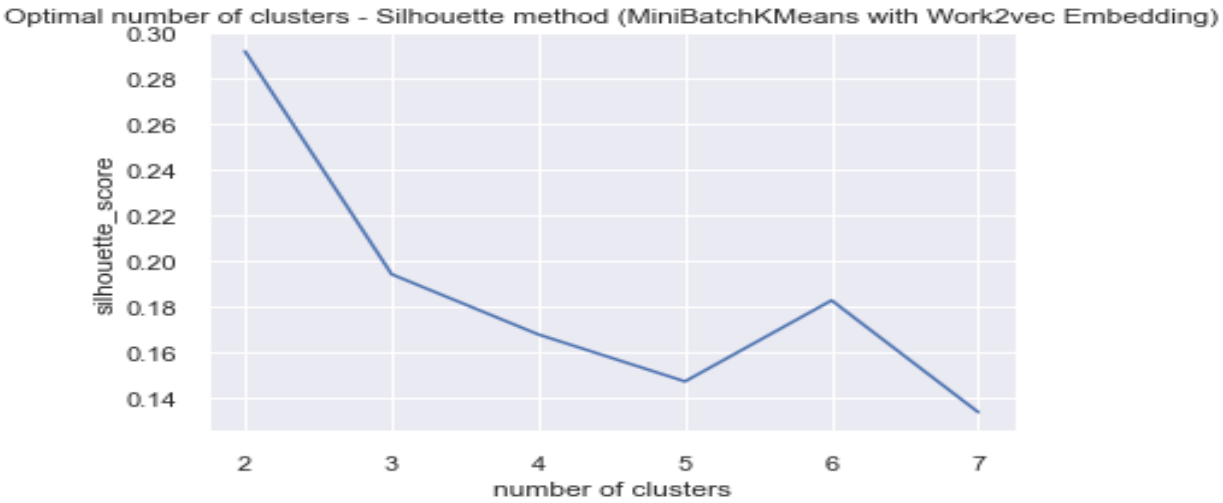

Figure 4.12: Optimal number of clusters - Silhouette method (Mini-Batch KMeans with Work2vec Embedding).

Table 4.4 shows that number of observations from using K=2, it can be observed that the number 0 has been assigned to cluster number 0, which contains true tweets about COVID-19, while the number 1 has been assigned to cluster number 1, which contains fake tweets about COVID-19. According to Table 4.4, the K-means algorithm clustered 189,224 tweets into cluster number 0 and 310,776 tweets into cluster number 1. Similarly, the Mini-Batch K-means algorithm clustered 219,573 tweets into cluster number 0 and 280,427 tweets into cluster number 1 in the given data.

Table 4.4: Number of observations per cluster (word2vec word embedding).

Model	Number of observations By Cluster
K-means	**{1: 310776, 0: 189224}**
mini-batch k-means	**{0: 219573, 1: 280427}**

It would be worthwhile to determine the extent of agreement between both algorithms in their tweet classification. One way to measure this is by using a confusion matrix, which is indexed in one dimension by the K-means clustering of the tweets and in the other by Mini-Batch K-means clustering. However, since we do not have true labels, the confusion matrix can only provide an indication of the level of agreement between the two clustering methods. Tables 4.5 show the crosstabulation generated by both the Mini-Batch K-means and K-means algorithms.

Table 4.5: Crosstabulation (word2vec word embedding).

		K-means	
	Clusters	**0**	**1**
mini-batch k-means	**0**	34,268	185,305
	1	154,956	125,471

In Table 4.5, we have a 2x2 crosstabulation.

The total of diagonal elements in the crosstabulation represents those items that the two clustering methods have put in the same class, which is 31.9% (34,268 + 125,471)/500,000 of all the tweets where both models agreed on the classification.

Table 4.6 presents a 2x2 confusion matrix for our models' performance on the labelled data discussed in Section 4.2. The summation of the diagonal elements within the matrix represents the count of accurately classified instances, while the rest of the values indicate the number of misclassified instances. The K-means model, which utilized Word2vec word embedding, achieved an accuracy of 71% on the labelled data. On the other hand, the Mini-Batch K-means model that also used word2vec word embedding achieved an accuracy of 40% on the labelled data. Mini-Batch K-Means yields outcomes that are generally only slight inferiority than those of K-means (Maharjan, 2018).

Table 4.6: Confusion Matrix (word2vec word embedding).

K-means								
		Actual				Actual	
	Word2Vec	fake	real		Word2Vec	fake	real
Predicted	fake	1,640	1,144	Predicted	fake	26%	18%
	real	1,420	2,216		real	22%	35%
Mini-Batch K-means							
		Actual				Actual	
	Word2Vec	fake	real		Word2Vec	fake	real
Predicted	fake	1,744	2,541	Predicted	fake	27%	40%
	real	1,316	819		real	20%	13%

The results from a 2x2 crosstabulation matrix involves analysing the relationships between two categorical variables. The matrix provides a summary of the counts or frequencies of observations falling into different categories of the variables. Table 4.7 presents a comprehensive 2x2 crosstabulation matrix pertaining to the labelled data discussed in section 4.2. The cumulative sum of the diagonal elements within the crosstabulation signifies the items that the two clustering methods have assigned to the same class, accounting for 10.1% (649/6420) of all the tweets where both models reached a consensus on the classification. This agreement rate serves as a valuable indicator of the reliability and consistency of the clustering techniques in terms of their classification decisions. However, it is important to note that this finding suggests a comparatively low level of agreement between the two clustering methods.

Further investigation of the off-diagonal elements of the matrix, which correspond to instances where the clustering methods exhibited discrepancies in their classifications, reveals a significant level of divergence or uncertainty between the models.

Table 4.7: Crosstabulation for the labelled data (word2vec word embedding).

		K-means	
	Clusters	**0**	**1**
mini-batch k-means	**0**	0	2,135
	1	3,636	649

The results from confusion matrix and crosstabulation were obtained using the K-means(n_clusters=2) and MiniBatchK-means(n_clusters=2) functions in Python. Default parameters were used for both K-means clustering and Mini-Batch K-means. The parameters for K-means used for all these experiments in this thesis is seen in Figure 4.13. These parameters collectively define the behaviour and settings for the K-means clustering algorithm.

```
{'algorithm': 'lloyd',
 'copy_x': True,
 'init': 'k-means++',
 'max_iter': 300,
 'n_clusters': 2,
 'n_init': 'warn',
 'random_state': None,
 'tol': 0.0001,
 'verbose': 0}
```

Figure 4.13: parameters for K-means.

The interpretation of these parameters is as follows:

'algorithm':

- 'algorithm': 'lloyd', refers to the algorithm used to compute the K-means clustering. It is also known as the standard K-means algorithm.
- 'copy_x': True, this parameter determines whether the algorithm should make a copy of the input data or not. In case a copy of the data was made, ensuring that the original data is not modified during the clustering process.

- 'init': 'k-means++', this parameter specifies the method used for initialization of the cluster centroids. 'k-means++' is a popular initialization method that intelligently selects the initial centroids to improve the convergence of the algorithm. In our case we have 2 clusters, so 2 centroid arrays were needed.
- 'max_iter': 300, this parameter defines the maximum number of iterations the algorithm will perform to converge to a solution. In our case 300 iterations were done, the maximum number of iterations allowed by the algorithm. The algorithm did not converge before reaching max iterations meaning it only terminated at the maximum iterations then current solution was returned.
- 'n_clusters': 2, this parameter delineates the desired quantity of clusters that the algorithm endeavours to discern within the dataset. In this case, the algorithm will aim to partition the data into two distinct clusters.
- 'n_init': 'warn', this parameter governs the frequency with which the algorithm is executed, employing distinct centroid seeds on each run. 'warn' suggests that the algorithm will display a warning message if the number of initializations is not provided explicitly.
- 'random_state': None, this parameter is used to seed the random number generator. Given that our specific configuration designates it as "None," distinct random states will be used for each iteration of the algorithm, thereby yielding potentially diverse solutions.
- 'tol': 0.0001, this parameter defines the tolerance or convergence criterion for the algorithm. If the change in the cluster centroids between two consecutive iterations is less than the specified tolerance, the algorithm is considered to have converged. Keeping this value low ensures that the entire feature space is scanned.
- 'verbose': 0, this parameter controls the verbosity of the algorithm's output. A value of 0 indicates that no additional output or information was displayed during the clustering process. This is the default setting, and it ensures that the algorithm ran silently without any extra messages or updates.

The above parameters can be briefly summarized using the following Python code snippet:

KMeans(*n_clusters=2*, *init='k-means++'*, *n_init='warn'*, *max_iter=300*, *tol=0.0001*, *verbose=0*, *random_state=None*, *copy_x=True*, *algorithm='lloyd'*)

The parameters for Mini-Batch K-means used for all these experiments in this paper is seen in figure 4.14. These parameters collectively define the behaviour and settings for the Mini-Batch K-means algorithm.

```
{'batch_size': 1024,
 'compute_labels': True,
 'init': 'k-means++',
 'init_size': None,
 'max_iter': 100,
 'max_no_improvement': 10,
 'n_clusters': 2,
 'n_init': 'warn',
 'random_state': None,
 'reassignment_ratio': 0.01,
 'tol': 0.0,
 'verbose': 0}
```

Figure 4.14: parameters for Mini-Batch K-means.

- 'batch_size': 1024, this parameter governs the magnitude of the mini-batches employed during the iterations of the algorithm. Mini-batches are subsets of the data used to update the cluster centroids. In our case, the batch size is set to 1024, meaning that 1024 data points was randomly sampled at each iteration.
- 'compute_labels': True, this parameter controls whether the algorithm should compute and assign labels to the input data based on the cluster centroids. In our specific case, with the parameter set to True, the algorithm will be configured to assign each data point to its closest centroid, subsequently providing the corresponding labels.
- 'init': 'k-means++', this parameter specifies the initialization method used to determine the initial cluster centroids. 'k-means++' is a popular initialization method that intelligently selects the initial centroids to improve the convergence of the algorithm. In our case we have 2 clusters, so 2 centroid arrays were needed.
- 'init_size': None, this parameter defines the size of the initial random sample used for centroids initialization. Since set to None, the entire dataset was used.

- 'max_iter': 100, this parameter establishes the upper limit on the number of iterations executed by the algorithm. After reaching this limit, the algorithm stopped iterating.

  This means the algorithm did not converge before reaching max iterations meaning it only terminated at the maximum iterations then current solution was returned.
- 'max_no_improvement': 10, this parameter controls the number of consecutive iterations without improvement in the clustering objective function.
- Fggg 'n_clusters': 2, this parameter delineates the desired quantity of clusters that the algorithm endeavours to discern within the dataset. In this case, the algorithm partitioned the data into two distinct clusters.
- 'n_init': 'warn', this parameter governs the frequency with which the algorithm is executed, employing distinct centroid seeds on each run. 'warn' suggests that the algorithm will display a warning message if the number of initializations is not provided explicitly.
- 'random_state': None, this parameter was employed as a seed for initializing the random number generator. Since set to None, a different random state was used for each execution of the algorithm, resulting in potentially different solutions.
- 'reassignment_ratio': 0.01, this parameter controls the percentage of data points that was reassigned to different clusters in each iteration. A higher value leads to more frequent reassignments and potentially faster convergence but can also result in less stable solutions.
- 'tol': 0.0, this parameter sets the tolerance or convergence criterion for the algorithm. A value of 0.0 indicates that the algorithm continued until the specified number of iterations was reached without checking for convergence.
- 'verbose': 0, this parameter controls the verbosity of the algorithm's output. Since the value was set to 0 it indicates that no additional output or information was displayed during the clustering process.

The above parameters can be briefly summarized using the following Python code snippet:

MiniBatchKMeans(*n_clusters=2, init='k-means++', max_iter=100, batch_size=1024, verbose=0, compute_labels=True, random_state=None, tol=0.0, max_no_improvement=10, init_size=None, n_init='warn', reassignment_ratio=0.01)*

Table 4.8 displays the first 20 tweets, where "clean_Text" refers to the cleaned tweets (unlabelled data) discussed in section 4.1, "cluster" represents the clustering done by K-means algorithm, and "cluster2" refers to the clustering done by Mini-Batch K-means algorithm. It can be observed that the first tweet was placed in cluster 0 by the K-means algorithm, while the same tweet was placed in cluster 1 by the mini-batch K-means algorithm. From the context of the tweets, we can infer that cluster 1 corresponds to fake tweets, while cluster 0 contains real tweets about COVID-19.

Table 4.8: Example of the tweets that have been clustered by K-means and mini-batch K-means using Word2vec word embedding.

Clean_Text	cluster	cluster2
brit reed caught kitten hospital confirmed cov...	1	0
impressive yes missing context question neurol...	1	0
going wager natural immunity letter rip pandem...	1	1
thanks castle theyre steep killing gamecovid y...	1	1
jayghai rio state immunization team excellent ...	0	1
plan continue wearing mask public mask mandate...	1	0
akeprofii day surveillance death specifically ...	0	0
dont remember bad player ducking smoke comment...	1	0
community health need helping stop spread covi...	1	0
didnt know beat covid uncertainty experience t...	1	0
happy covid shelter place day smoking collabor...	1	1
desi hirley oanguy aulril avid reeves damg uke...	1	1
health department ‘ fast loose covid profit ro...	0	1
covid increasing simple protect try stay home ...	1	0
administration confirmed making second covid b...	0	1
thanks partner positive met literally read mak...	1	0
accidental special checkers – evidence linking...	0	1
getting easier simply visit nearest vaccinatio...	0	0
important covid study infectious natural immu...	0	0
pacific islander news june audio language comm...	0	1

From the unlabelled data we can extract the most common words used in each cluster to identify the patterns in language used for fake tweets and real tweets about COVID-19. To begin with, we will examine cluster number 0 from the K-means algorithm. As mentioned in Table 4.4, this cluster contains a total of 189,224 tweets, and we can extract the most frequently occurring words from them. According to Figures 4.15 and 4.16, some of the common words in this cluster include "new," "health," "analytics," "data," and so on.

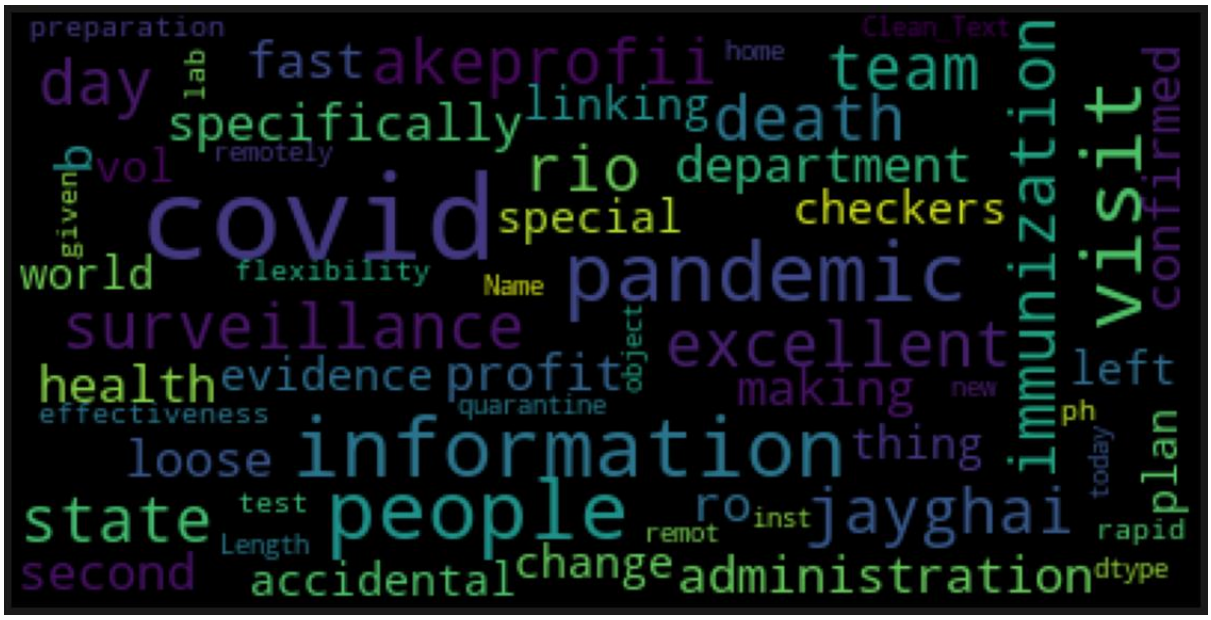

Figure 4.15: Word cloud for COVID-19 Tweets from cluster number 0 (real)from the K-means algorithm.

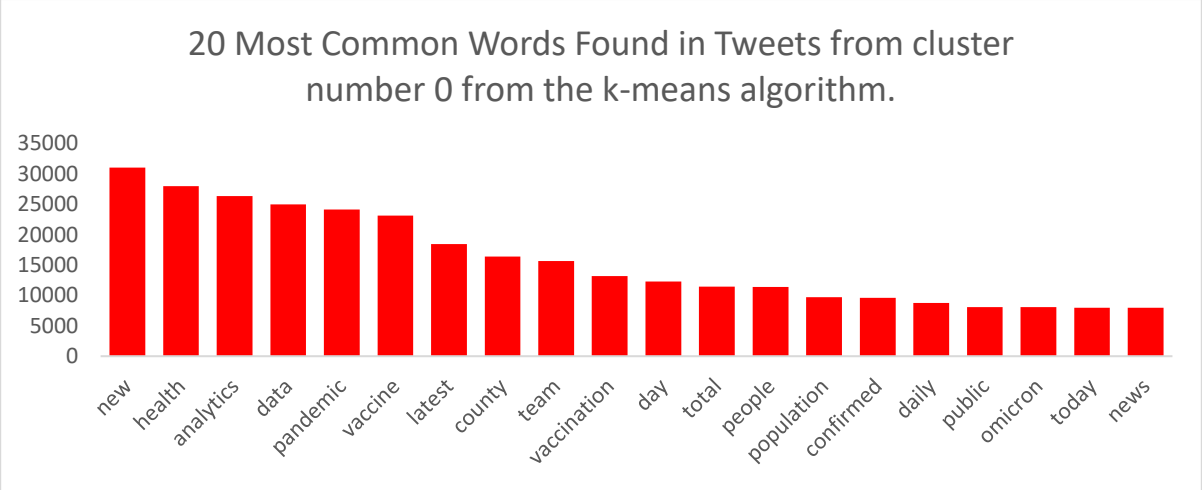

Figure 4.16: 20 Most Common Words Found in Tweets from cluster number 0 (real) from the K-means algorithm.

Next, let's focus on cluster number 1 from the K-means algorithm. According to Table 4.4, this cluster contains a total of 310,776 tweets, and we can extract the frequently occurring words from them. As illustrated in Table 4.4, Figures 4.17 and 4.18 exhibit that the most frequently occurring terms in this cluster are "people," "time," "vaccine," and so on.

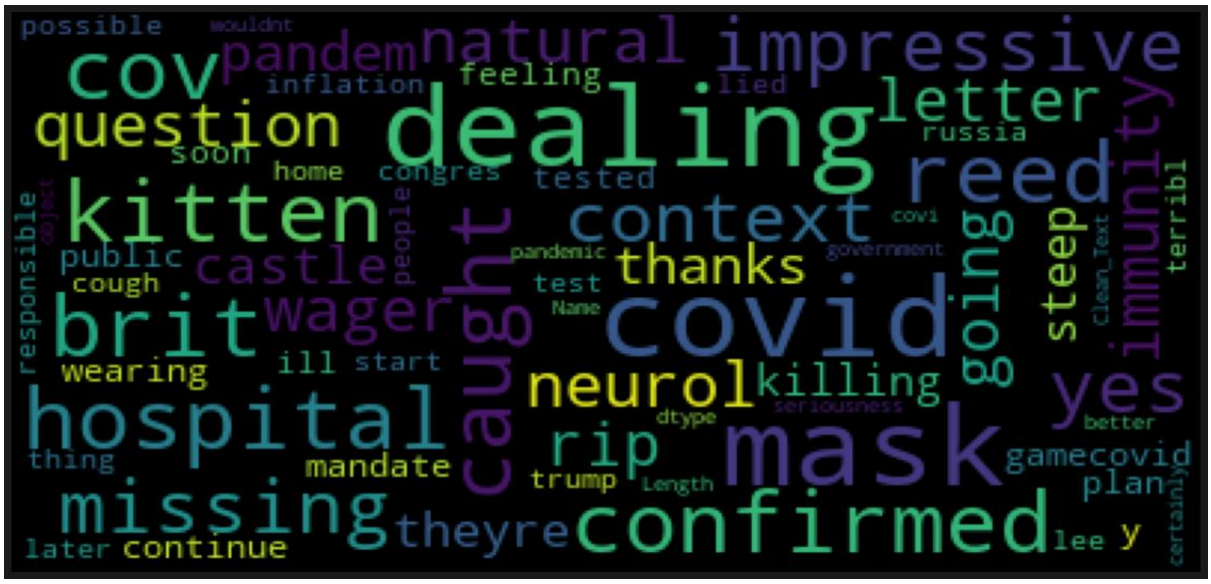

Figure 4.17: Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the K-means algorithm.

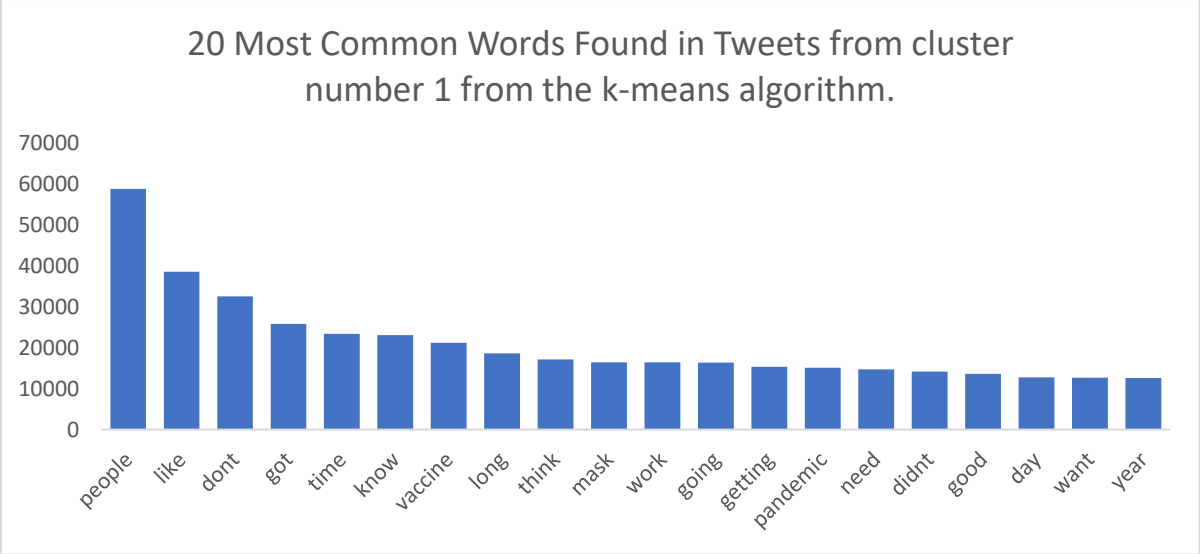

Figure 4.18: 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the K-means algorithm.

Turning our attention to cluster number 0 from the Mini-Batch K-means algorithm, as indicated in Table 4.4, this cluster encompasses 219,573 tweets, from which we can extract the most used words. Figures 4.19 and 4.20 reveal that prevalent terms in this cluster encompass "people," "vaccine," "test," and so forth.

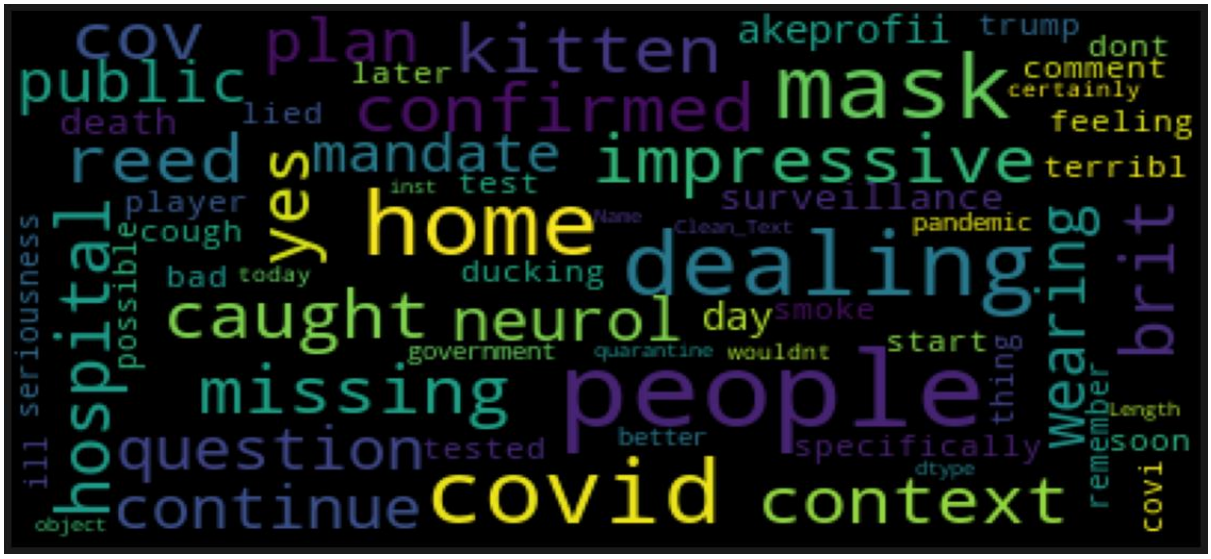

Figure 4.19: Word cloud for COVID-19 Tweets from cluster number 0 (real) from the mini-batch K-means algorithm.

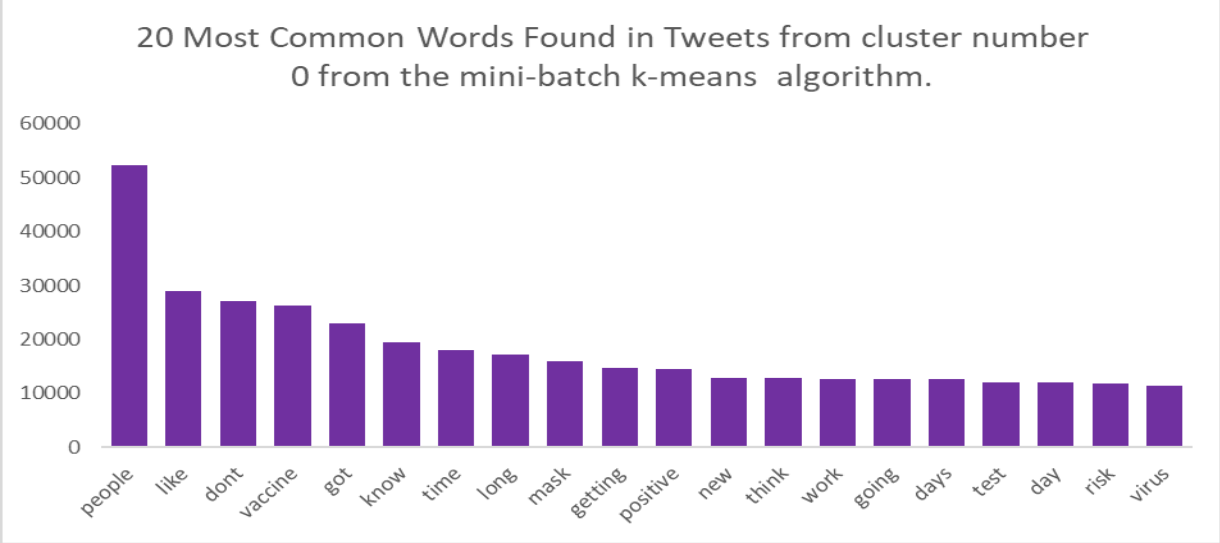

Figure 4.20: 20 Most Common Words Found in Tweets from cluster number 0 (real) from the mini-batch K-means algorithm.

Next, we shall proceed to analyse the cluster 1 extracted from the Mini-Batch K-means algorithm. Table 4.4 indicates that this cluster contains a total of 280,427 tweets, from which we can extract the frequently occurring words. As illustrated in Figures 4.21 and 4.22, common words in this cluster include "pandemic," "people," "report," "vaccine," and so on.

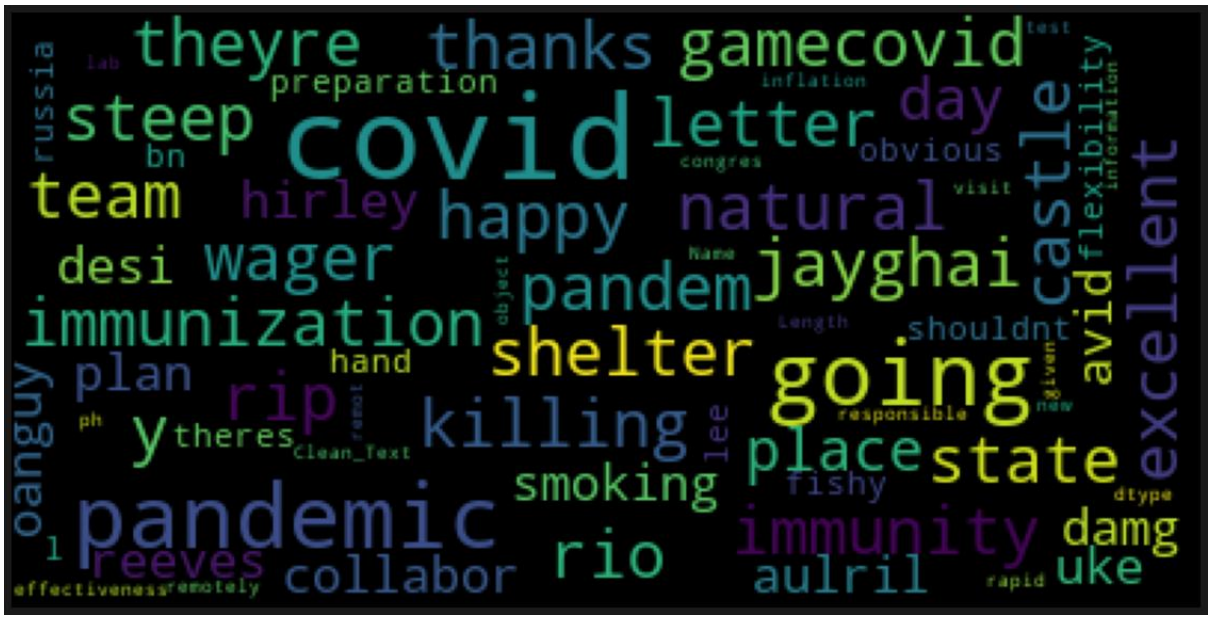

Figure 4.21: Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

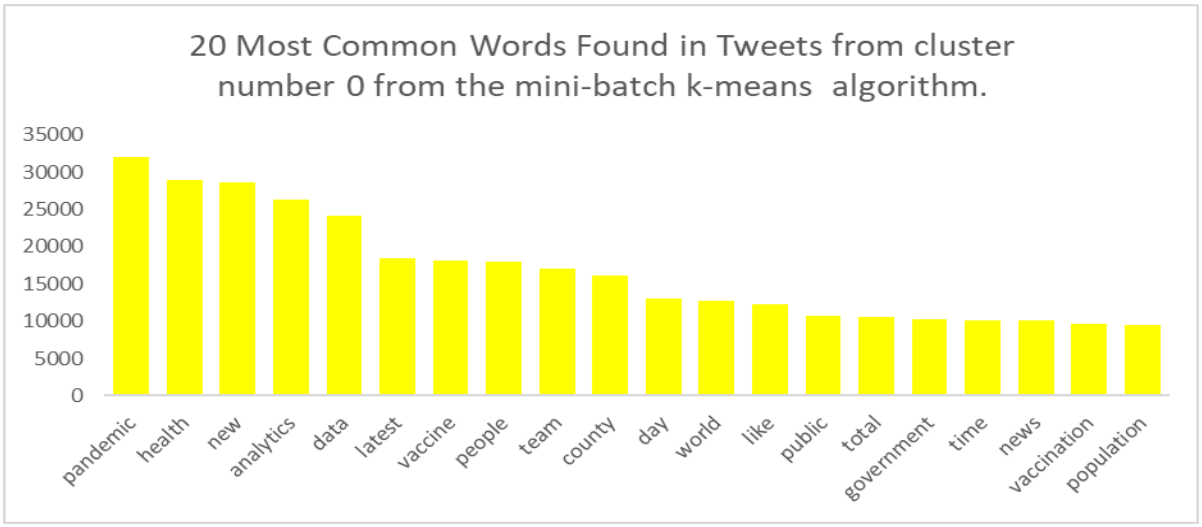

Figure 4.22: 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

In order to visualize the different clusters, a dimensionality reduction was performed on the resulting clustered unlabelled data. We chose to analyse the two clusters (fake and real COVID-19 tweets) obtained from the K-means algorithm.

To perform the dimensionality reduction, we used t-SNE and Truncated SVD methods. Figure 4.23 shows the two-dimensional representation of the clusters formed by the K-means algorithm. The figure on the left is a visualization of the t-SNE dimensionality reduction method representation of the clusters, while the figure on the right is Truncated SVD dimensionality reduction visualization. The blue cluster represents cluster 0, while the orange cluster represents cluster 1. The cluster in the sub-figure on the right is pear-shaped, while the sub-figure on the left is more oval-shaped. However, t-SNE did not do a good job in visualizing the different clusters in two dimensions, while Truncated SVD did a better job in clearly showing the two clusters. From the figure on the right, one can easily distinguish cluster 0 from cluster 1.

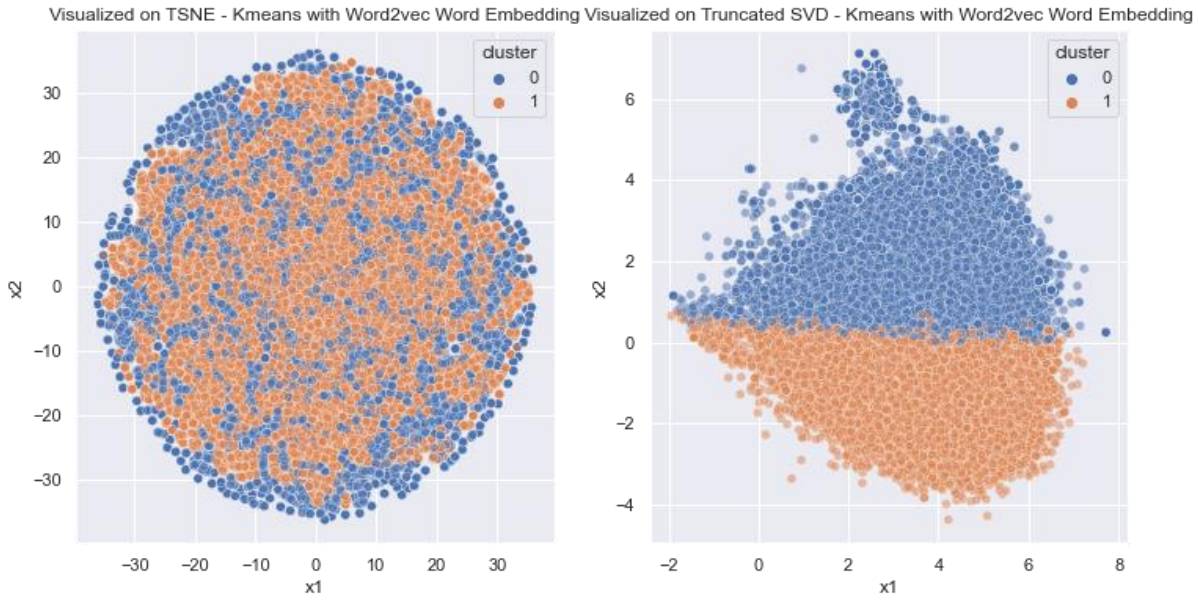

Figure 4.23: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization - K-means with Word2vec Word Embedding.

In Figure 4.24, we show a visual representation that illustrates the clusters generated by the Mini-Batch K-means algorithm is presented in two dimensions. The left figure depicts a cluster representation using the t-SNE dimensionality reduction technique, while the right figure represents it via Truncated SVD dimensionality reduction. Cluster 0 and Cluster 1 are depicted in blue and orange, respectively. The sub-figure on the right illustrates a pear-shaped cluster, while the left sub-figure exhibits an oval-shaped cluster.

Notably, Truncated SVD outperforms t-SNE in visualizing the distinct clusters in two dimensions, as evident from the right figure where both clusters are clearly visible.

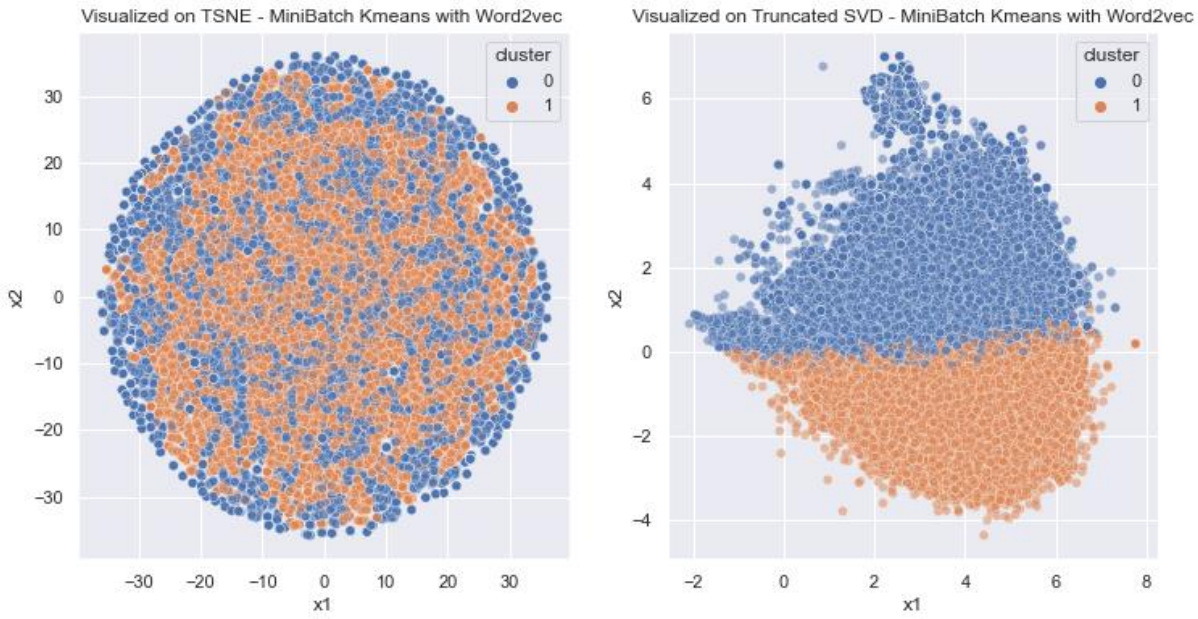

Figure 4.24: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization – Mini Batch K-means with Word2vec Word Embedding.

Truncated SVD on both figures (Figure 4.23 & Figure 4.24) look very similar. This trend is also seen in t-SNE visualisation on both figures (Figure 4.23 & Figure 4.24). We present visualizations of the clusters identified in the labelled data discussed in section 4.2. Figure 4.25 illustrates the two-dimensional representations of the clusters obtained using the K-means algorithm with Word2vec Word Embedding. The left figure showcases the t-SNE dimensionality reduction method visualization, while the right figure displays the visualization based on Truncated SVD dimensionality reduction.

In the visualizations, the blue cluster corresponds to cluster 0, while the orange cluster represents cluster 1. Notably, the cluster depicted in the right sub-figure exhibits a triangle shaped pattern, whereas the left sub-figure displays a more web like shape. Both t-SNE and Truncated SVD techniques effectively capture the distinct clusters in the two-dimensional space. The visualizations allow for clear differentiation between cluster 0 and cluster 1, facilitating easy identification of the two clusters.

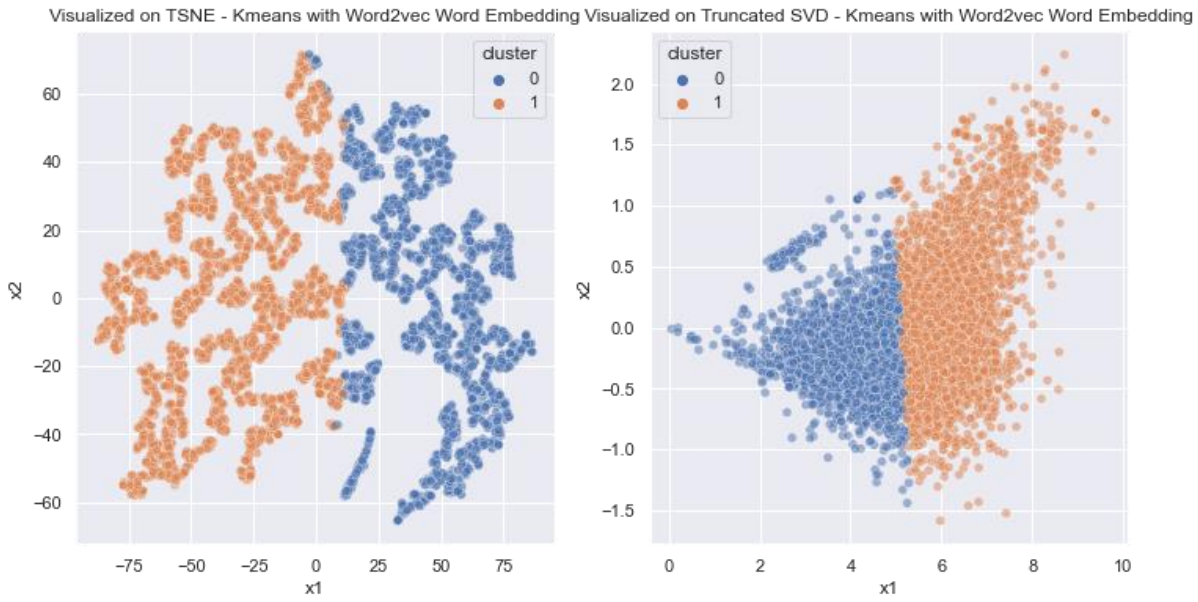

Figure 4.25: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization - K-means with Word2vec Word Embedding (labelled data).

In Figure 4.25, we present a graphical depiction illustrating the clusters generated by the K-means algorithm with Word2vec Word Embedding on the labelled data discussed in section 4.2. The left figure showcases the utilization of the t-SNE dimensionality reduction technique to represent the clusters, while the right figure employs the Truncated SVD dimensionality reduction method for visualization. The clusters are identified as Cluster 0 and Cluster 1, displayed in blue and orange colours, respectively.

Upon examining the visualizations, it becomes apparent that the cluster portrayed in the right sub-figure exhibits a distinct triangle shaped pattern, whereas the left sub-figure displays a more web like shape. Both the t-SNE and Truncated SVD techniques effectively capture the distinct clusters in a two-dimensional space. These visualizations serve as valuable tools for distinguishing between Cluster 0 and Cluster 1, facilitating a clear identification of the individual clusters. In Figure 4.26, a visual representation is provided, delineating the clusters derived through the application of the Mini-Batch K-means algorithm utilizing Word2vec Word Embedding on the annotated data expounded upon in Section 4.2.

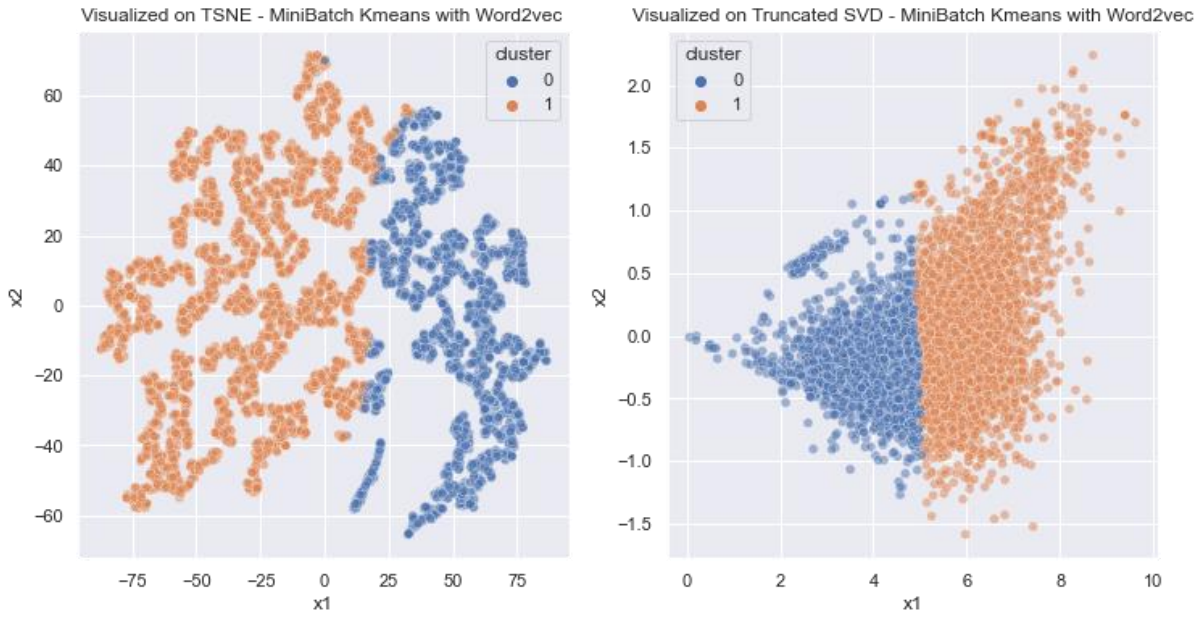

Figure 4.26: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization - Mini Batch K-means with Word2vec Word Embedding (labelled data).

Truncated SVD on both figures (Figure 4.25 & Figure 4.26) look very similar. This trend is also seen in t-SNE visualisation on both figures (Figure 4.25 & Figure 4.26).

4.3.2 PERFORMANCE ANALYSIS WITH BERT WORD EMBEDDING.

In Figure 4.27, the illustration portrays the optimal determination of the number of clusters through the Silhouette method. Notably, the maximization of the silhouette score at K = 2 signifies the optimal segmentation into two clusters for the application of K-means clustering employing a BERT embedding. Consequently, the segmentation into two clusters shall be pursued.

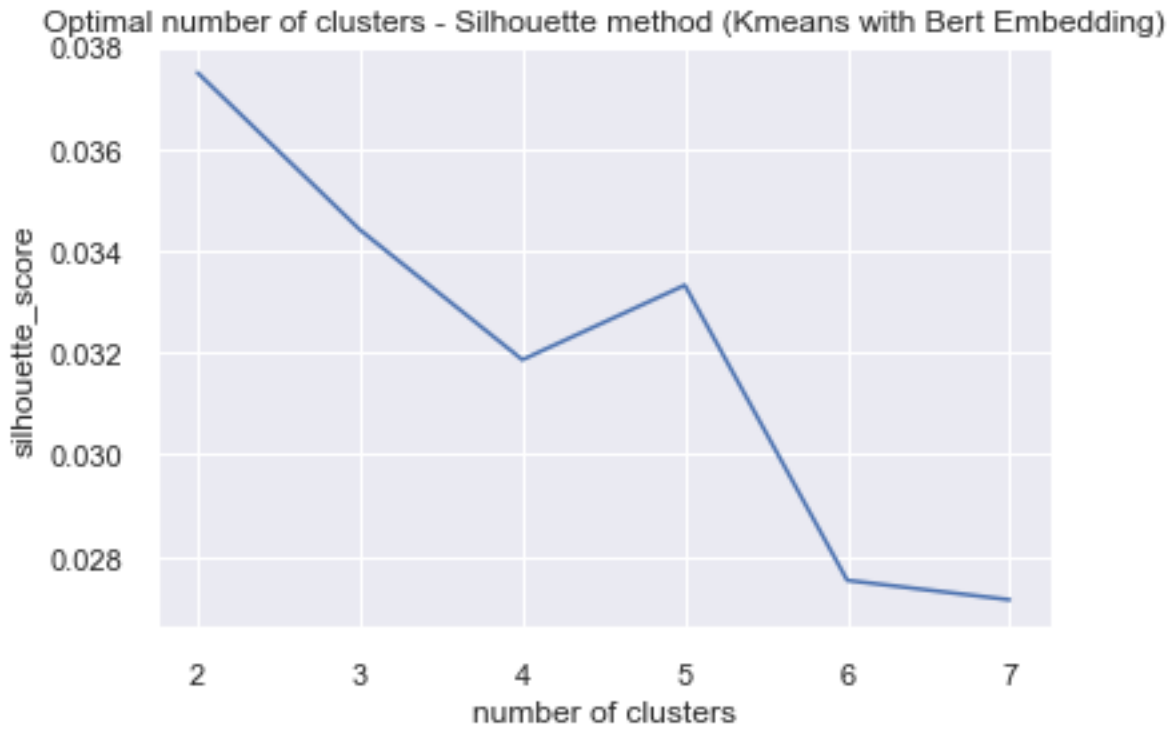

Figure 4.27: Optimal number of clusters - Silhouette method (K-means with BERT Embedding).

Within Figure 4.28, the graphic representation delineates the optimal determination of cluster quantity via the Silhouette method. Significantly, the achievement of maximal silhouette score at K = 2 indicates the optimal partitioning into two clusters when employing the Mini-Batch K-means clustering approach with a BERT embedding. Consequently, the pursuit of segmentation into two clusters is warranted.

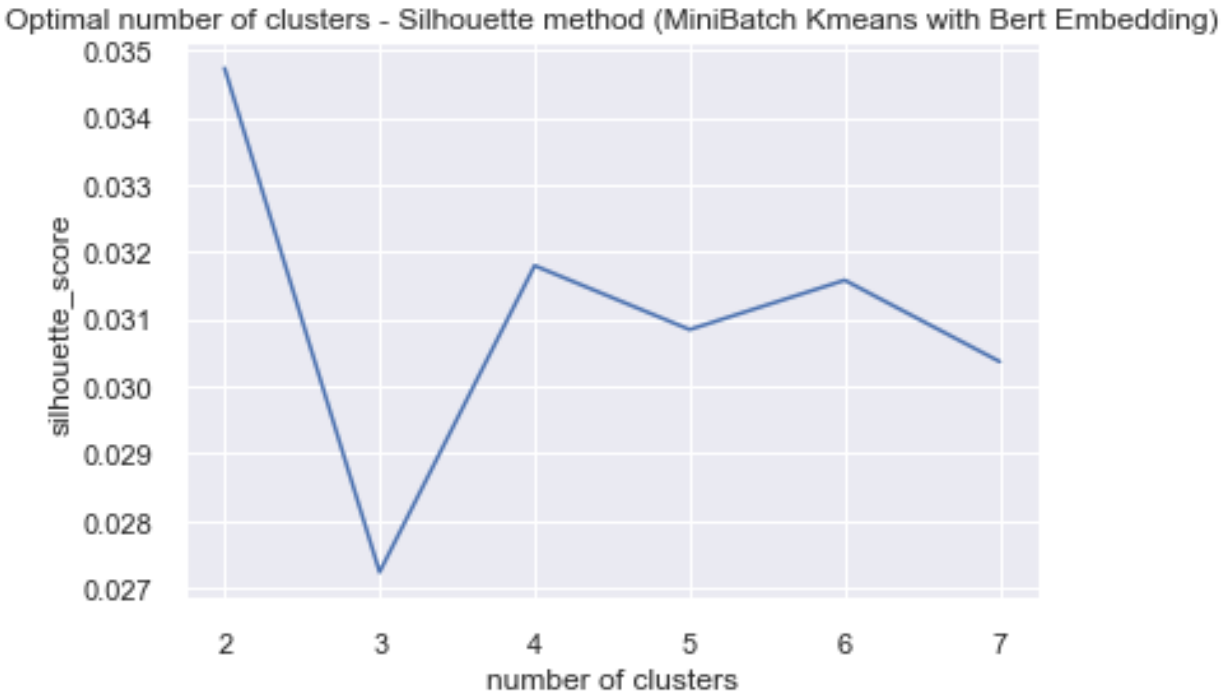

Figure 4.28: Optimal number of clusters - Silhouette method (Mini-Batch K-means with BERT Embedding).

Using K=2 one observes that cluster number 0, encompassing authentic tweets about COVID-19, is designated as 0, while cluster number 1, encompassing fraudulent tweets about COVID-19, is designated as 1. As indicated in Table 4.9, the K-means algorithm assigned 240,286 tweets to cluster number 0 and 259,714 tweets to cluster number 1. Analogously, in the provided dataset, the mini-batch K-means algorithm classified 254,397 tweets into cluster number 0 and 245,603 tweets into cluster number 1.

Table 4.9: Number of observations per cluster (Bert word embedding).

Model	Number of observations By Cluster
K-means	{1: 259714, 0: 240286}
mini-batch k-means	{0: 254397, 1: 245603}

Examining the overlap in the classification conducted by both algorithms is of great value. An optimal method to assess this would be to employ a confusion matrix, which could be indexed in one dimension by the K-means clustering and in the other by the Mini-Batch K-means clustering. However, since there are no actual labels available, the confusion matrix or crosstabulation would only reveal the level of agreement between the two clustering techniques. Table 4.10 presents the crosstabulations produced by the Mini-Batch K-means and K-means algorithms utilizing Bert word embedding.

Table 4.10: Crosstabulation (Bert word embedding).

		K-means	
	Clusters	**0**	**1**
mini-batch k-means	**0**	132073	122324
	1	127641	117962

Table 4.10 presents a 2x2 crosstabulation where the sum of diagonal elements represents the number of items classified identically by both clustering methods. These values correspond to 50.01% (132,073 + 117,962) of all the tweets, indicating that the two models agreed on the classification for this proportion of the dataset. In Section 4.2, we discussed the labelled data, and Table 4.11 displays a 2x2 confusion matrix representing the performance of our model on this data. The summation of the diagonal elements within the matrix represents the count of accurately classified instances, while the rest of the values indicate the number of misclassified instances. The K-means model, which employed Bert word embedding, achieved 76% accuracy on the labelled data.

In contrast, the Mini-Batch K-means model that employed Bert word embedding achieved 47% accuracy on the labelled data. Notably, Mini-Batch K-means performed less accurately than K-means.

Table 4.11: Confusion Matrix (Bert word embedding).

K-means							
		Actual				Actual	
	Bert	fake	real		Bert	fake	real
Predicted	fake	1,906	395	Predicted	fake	30%	6%
	real	1,154	2,965		real	18%	46%
Mini-Batch K-means							
		Actual				Actual	
	Bert	fake	real		Bert	fake	real
Predicted	fake	1,670	1,986	Predicted	fake	26%	31%
	real	1,390	1,374		real	22%	21%

The 2x2 crosstabulation matrix provides a concise overview of the frequency or count distribution of observations across different categories of the variables. Table 4.12 presents a comprehensive 2x2 crosstabulation matrix that pertains to the labelled data discussed in section 4.2. The cumulative sum of the diagonal elements in the matrix represents the items that both clustering methods assigned to the same class, amounting to 58.4% ((2106+1643)/6420) of all the tweets where both models reached a consensus on the classification. This agreement rate serves as a significant indicator of the reliability and consistency of the clustering techniques in terms of their classification decisions.

It is imperative to acknowledge, however, that this finding suggests a moderate level of agreement between the two clustering methods. Further examination of the off-diagonal elements of the matrix, which correspond to instances where discrepancies emerged in the classifications between the clustering methods, reveals a moderate degree of divergence or uncertainty between the models.

Table 4.12: Crosstabulation for the labelled data (Bert word embedding).

		K-means	
	Clusters	0	1
mini-batch k-means	0	2106	658
	1	2013	1643

Table 4.13 displays the first 20 tweets, corpus refers to the cleaned tweets (unlabelled data) discussed in section 4.1, which were classified by the K-means and Mini-Batch K-means algorithms into "cluster" and "cluster2", respectively. The initial tweet was classified into cluster 0 by the K-means algorithm, while the Mini-Batch K-means algorithm classified it into cluster 1. Upon reviewing the content of the tweets, we can infer that cluster 1 corresponds to fake tweets, while cluster 0 consists of real tweets related to COVID-19. This highlights the difference in classification between the two algorithms, with the K-means algorithm having a better accuracy rate on the labelled data as discussed in section 4.2.

Table 4.13: Example of the tweets that have been clustered by K-means and mini-batch K-means using BERT word embedding.

corpus	cluster	cluster2
brit reed caught kitten hospital confirmed cov...	1	0
impressive yes missing context question neurol...	0	1
going wager natural immunity letter rip pandem...	0	0
thanks castle theyre steep killing gamecovid y...	0	1
jayghai rio state immunization team excellent ...	1	0
plan continue wearing mask public mask mandate...	1	0
akeprofii day surveillance death specifically ...	1	1
dont remember bad player ducking smoke comment...	0	0
community health need helping stop spread covi...	0	0
didnt know beat covid uncertainty experience t...	0	0
happy covid shelter place day smoking collabor...	1	0
desi hirley oanguy aulril avid reeves damg uke...	1	0
health department ‘ fast loose covid profit ro...	1	1
covid increasing simple protect try stay home ...	0	0
administration confirmed making second covid b...	1	1
thanks partner positive met literally read mak...	1	1
accidental special checkers – evidence linking...	0	0
getting easier simply visit nearest vaccinatio...	1	0
important covid study infectious natural immu...	1	0
pacific islander news june audio language comm...	1	0

We can analyse the prevalent words in each cluster to gain insights into the common language used in real and fake tweets related to COVID-19. Let us begin by focusing on cluster 0 of the K-means algorithm. According to Table 4.9, this cluster contains 240,286 tweets, and we can extract the most commonly occurring words from them. Based on Figures 4.29 and 4.30, some of the frequently used words in this cluster are "people," "don't," "like," "pandemic," and so on.

Figure 4.29: Word cloud for COVID-19 Tweets from cluster number 0 (real) from the K-means algorithm.

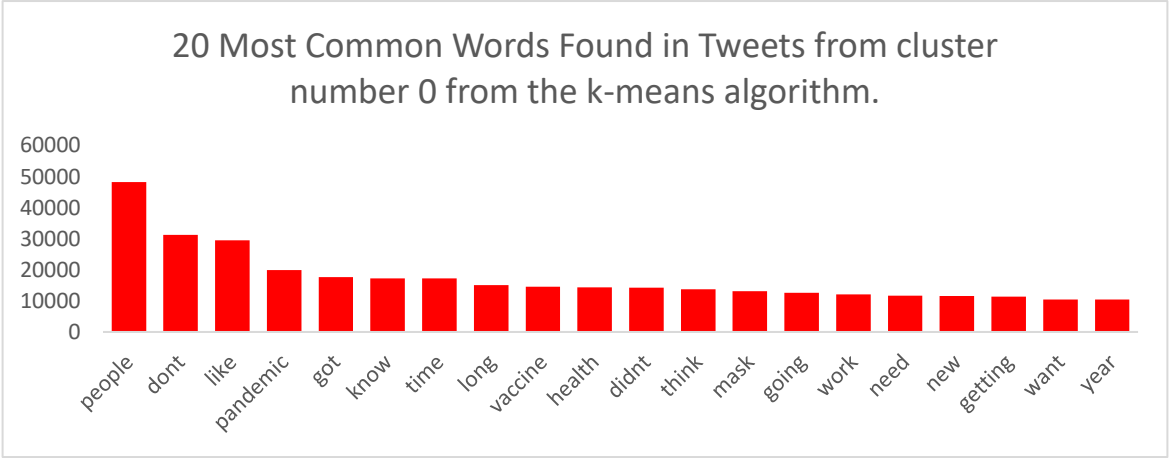

Figure 4.30: 20 Most Common Words Found in Tweets from cluster number 0 (real) from the K-means algorithm.

Moving on to cluster number 1 identified by the K-means algorithm, as shown in Table 4.8, it encompasses 259,714 tweets. By extracting the most common words from this cluster, we can gain insights into the language frequently used in fake tweets about COVID-19. As illustrated in Figures 4.31 and 4.32, common words in this cluster include "new," "vaccine," "analytics," and others.

Figure 4.31: Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the K-means algorithm.

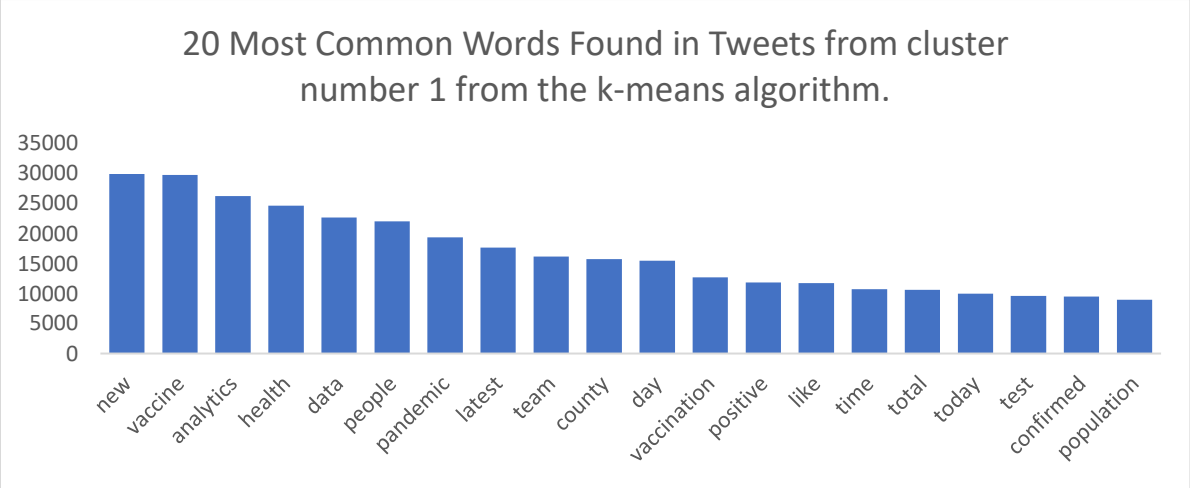

Figure 4.32: 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the K-means algorithm.

Let's now shift our attention to cluster 0 obtained from the mini-batch K-means algorithm, which consists of 254,397 tweets. Through an analysis of the prevalent terms within this cluster, as exemplified in Figures 4.33 and 4.34, a deeper understanding of the commonly employed language in these tweets can be obtained. It is noteworthy that words such as "vaccine", "people", and "health" frequently appear in this cluster.

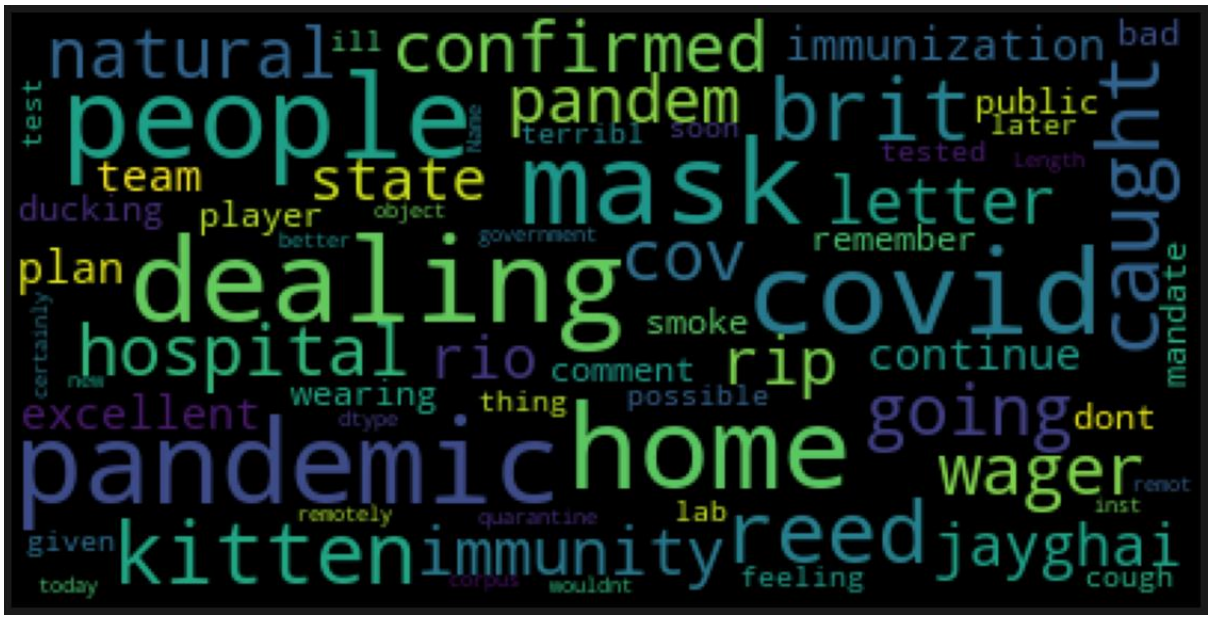

Figure 4.33: Word cloud for COVID-19 Tweets from cluster number 0 (real) from the Mini-Batch K-means algorithm.

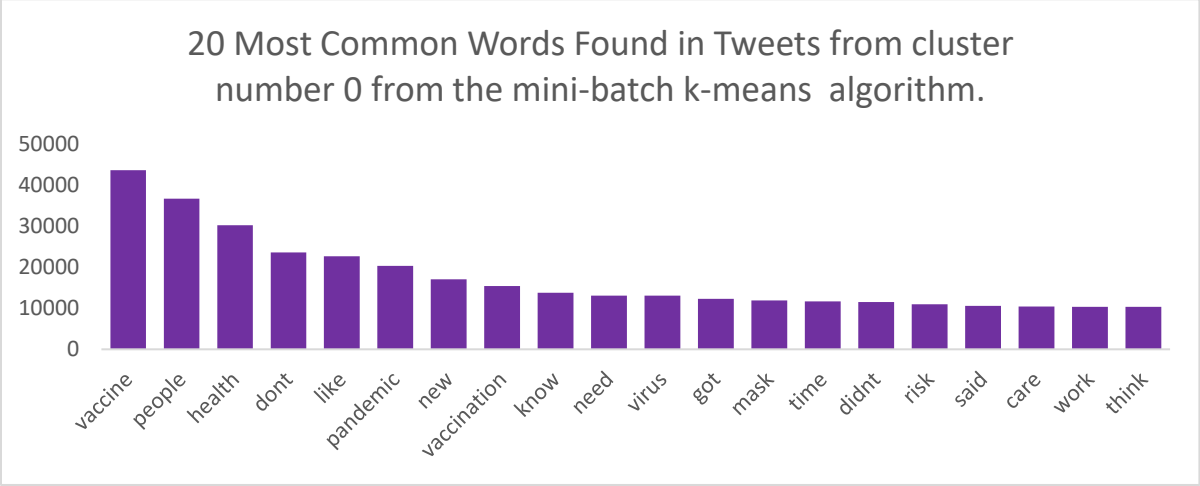

Figure 4.34: 20 Most Common Words Found in Tweets from cluster number 0 (real) from the Mini-Batch K-means algorithm.

Next, let's take a closer look at cluster number 1 from the mini-batch K-means algorithm. Table 4.9 indicates that this cluster contains a total of 245,603 tweets. We can extract the most commonly occurring words from these tweets to gain insights into the language commonly used in this cluster. Figures 4.35 and 4.36 reveal that words such as “people”, “analytics”, “new”, and “data” are among the most frequent words in this cluster.

Figure 4.35: Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

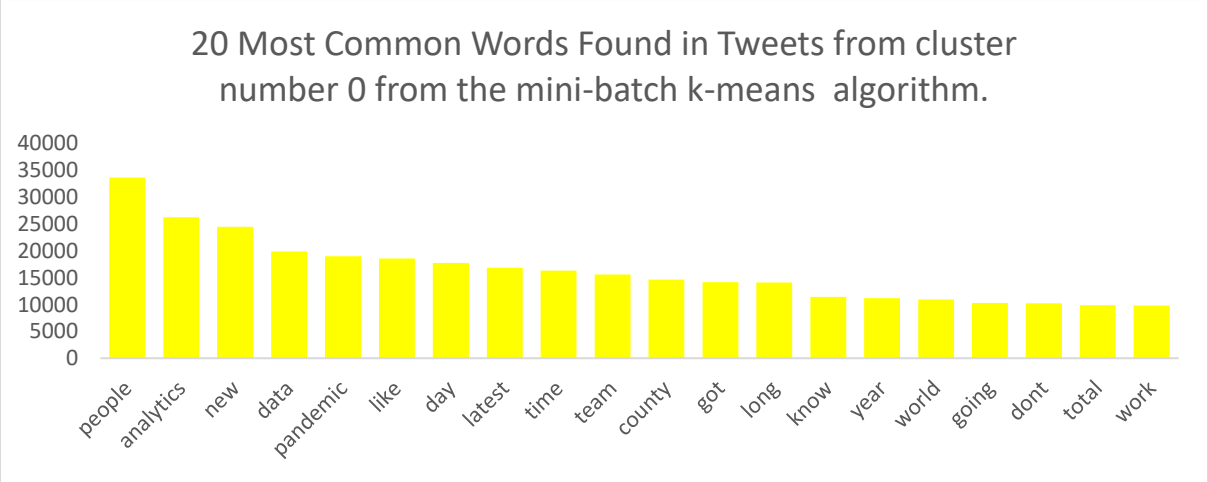

Figure 4.36: 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

To visualize the resulting data clusters, we utilized dimensionality reduction techniques such as t-SNE and Truncated SVD. We selected the of clusters as 2 (real and fake COVID-19 tweets) for the cluster analysis. Figure 4.37 provides a graphical depiction of the clusters produced by the K-means algorithm, visually representing their distribution in a two-dimensional space.

This visualization offers insights into the spatial arrangement and relationships among the generated clusters. The visualization on the left side shows the t-SNE representation of the clusters, while the visualization on the right side shows the Truncated SVD representation. The cluster in the sub-figure on the right is pear-shaped, while the sub-figure on the left is more oval-shaped. The blue colour cluster corresponds to cluster 0, while the orange colour cluster corresponds to cluster 1. However, t-SNE did not effectively visualize the different clusters in two dimensions, while Truncated SVD provided a better visualization of the clusters. The figure on the right clearly shows the distinction between cluster 0 and cluster 1.

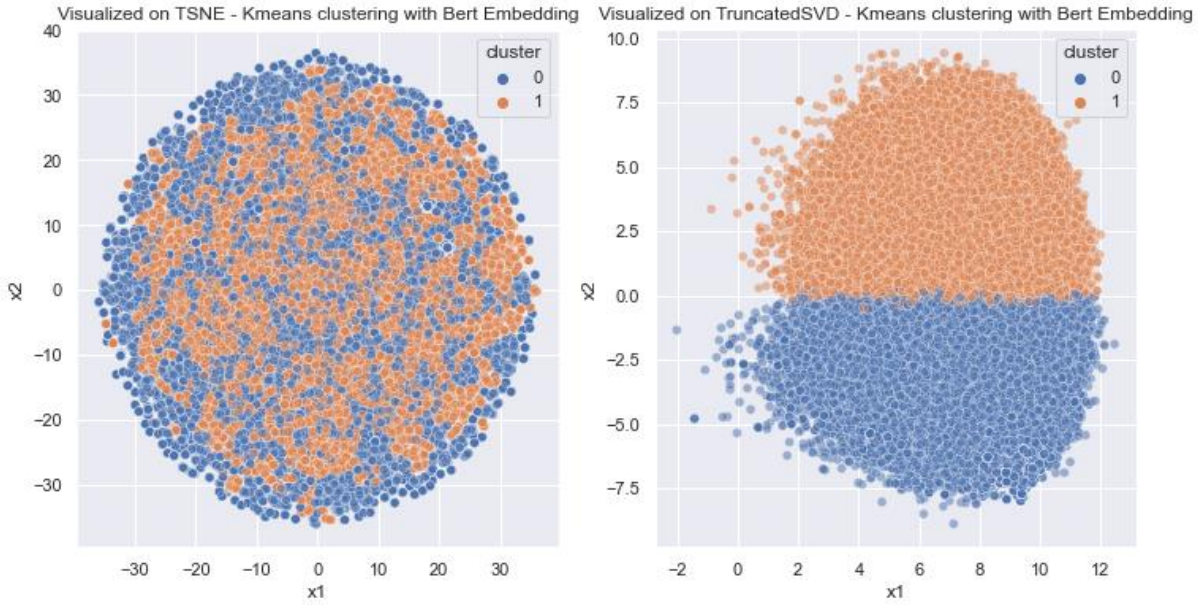

Figure 4.37 t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization – K-means with BERT Word Embedding.

Figure 4.38 provides a graphical depiction of the clusters produced by the Mini-Batch K-means algorithm, illustrating their spatial distribution in a two-dimensional space. The left image utilizes the t-SNE dimensionality reduction technique to represent the clusters, while the right image employs Truncated SVD. Cluster 0 and Cluster 1 are represented by blue and orange colours, respectively.

The left sub-figure displays an oval-shaped cluster, while the right sub-figure shows a pear-shaped cluster. Notably, Truncated SVD provides better visualization of the separate clusters in two dimensions, as seen in the right image where both clusters are clearly distinguishable.

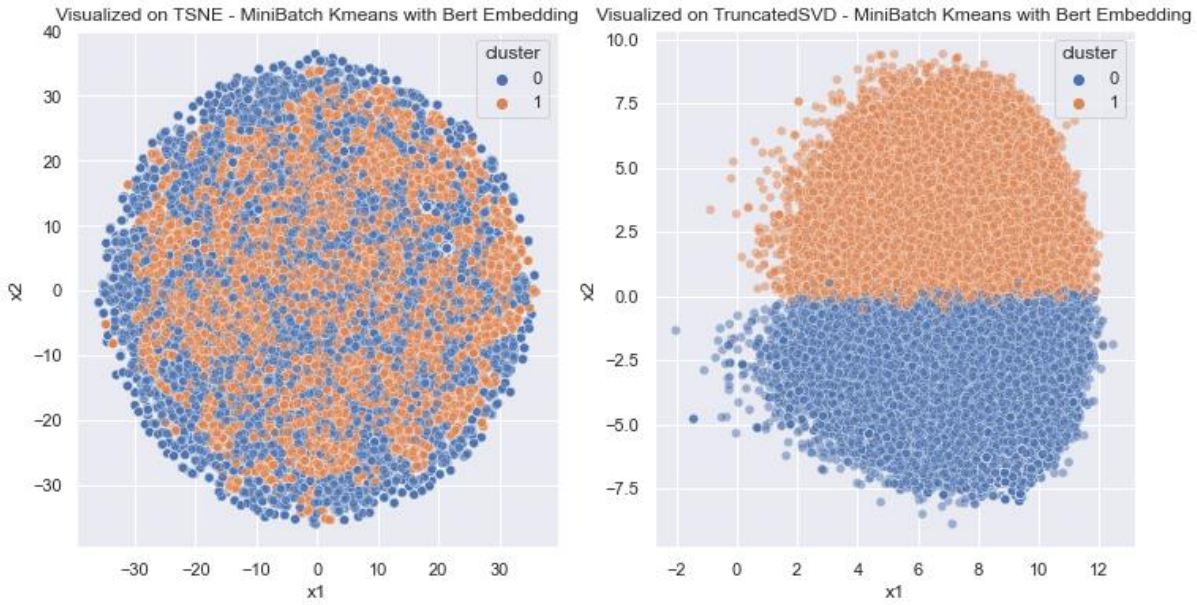

Figure 4.38 t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization – Mini Batch K-means with BERT Word Embedding.

We now present visualizations of the clusters identified in the labelled data discussed in section 4.2. Figure 4.39 illustrates the two-dimensional representations of the clusters obtained using the K-means algorithm with Bert word embedding. The left figure presents the visualization based on the t-SNE dimensionality reduction method, while the right figure depicts the visualization derived from the Truncated SVD dimensionality reduction technique. In these visualizations, the blue cluster corresponds to cluster 0, while the orange cluster represents cluster 1. It is worth noting that the cluster portrayed in the right sub-figure exhibits a distinctive pear-shaped pattern, while the left sub-figure displays a more intricate web-like shape. Both the t-SNE and Truncated SVD techniques effectively capture the unique characteristics of the clusters within the two-dimensional space. The visualizations allow for clear differentiation between cluster 0 and cluster 1, facilitating easy identification of the two clusters.

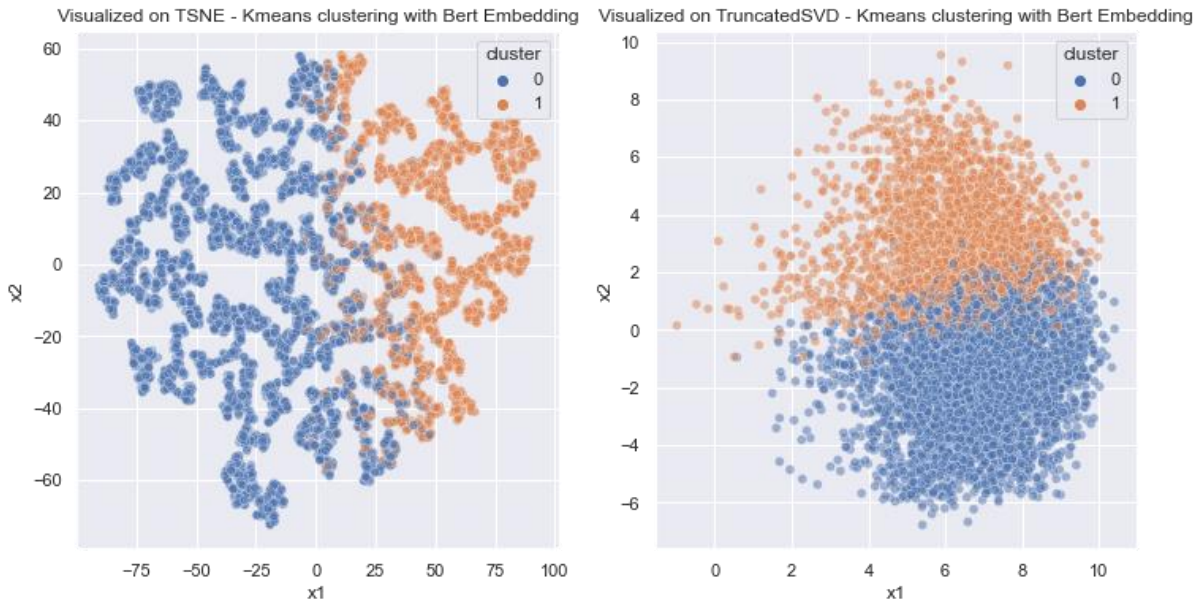

Figure 4.39 t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization – K-means with BERT Word Embedding (labelled data).

In Figure 4.40, we present a graphical depiction illustrating the clusters generated by the Mini-Batch K-means algorithm with Bert word embedding on the labelled data discussed in section 4.2. The left figure showcases the utilization of the t-SNE dimensionality reduction technique to represent the clusters, while the right figure employs the Truncated SVD dimensionality reduction method for visualization. The clusters are labelled as Cluster 0 and Cluster 1, depicted in blue and orange colours, respectively.

Upon careful examination of the visualizations, it becomes evident that the cluster depicted in the right sub-figure exhibits a distinctive pear-like shaped pattern, whereas the left sub-figure displays a more intricate web-like shape. However, it is notable that both the t-SNE and Truncated SVD techniques struggle to effectively capture the distinct clusters in the two-dimensional space. Regrettably, these visualizations fail to offer valuable means for a distinct differentiation between Cluster 0 and Cluster 1. These findings suggest Mini-Batch K-means algorithm with failed to accurately differentiate the identified clusters in the given labelled data. The visual representations inadequately depict the inherent characteristics and boundaries of the clusters, thus constraining their efficacy in effectively discerning between the two clusters.

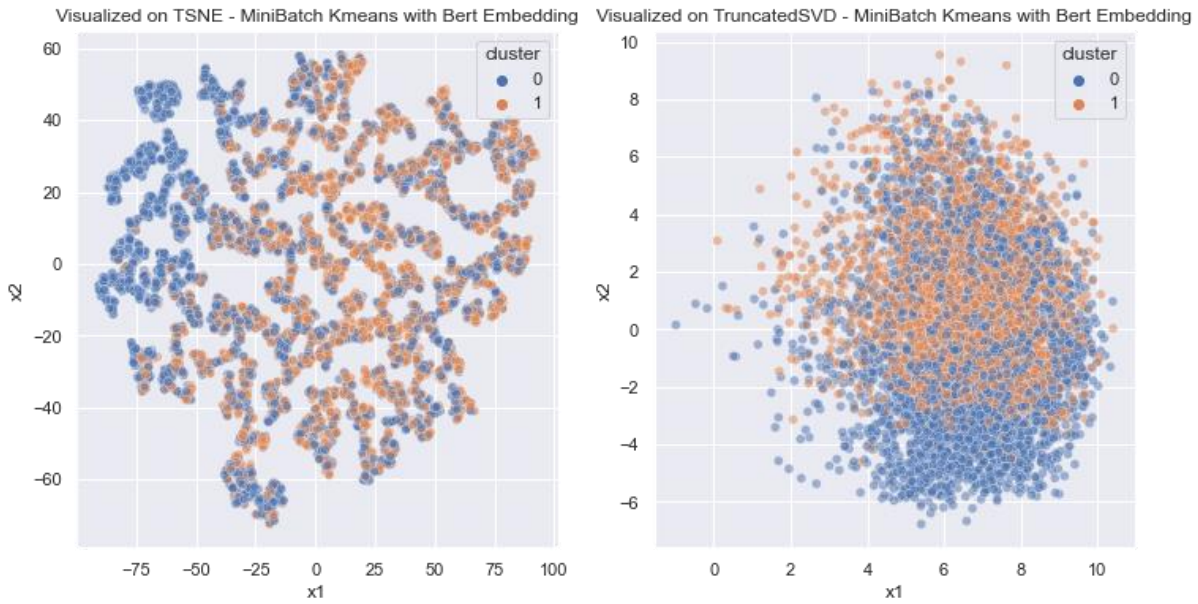

Figure 4.40 t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization - Mini Batch K-means with BERT Word Embedding (labelled data).

4.3.3 PERFORMANCE ANALYSIS WITH TF-IDF WORD EMBEDDING.

In Figure 4.41, the illustration portrays the optimal determination of the number of clusters through the Silhouette method. Notably, the maximization of the silhouette score at K = 7 signifies the optimal segmentation into seven clusters for the application of K-means clustering employing a TF-IDF embedding. As previously explicated, a silhouette score of one implies that each data point is unlikely to be assigned to another cluster, while a score approximating zero suggests the ease with which each data point could be reassigned to another cluster. Inspection of Figure 4.41 reveals a silhouette score in close proximity to zero, indicating the potential ease of reassignment of each data point to another cluster. It is imperative to exercise caution in placing unwavering reliance on silhouette scores, particularly when exhibiting low values, as they may not reliably discern the optimal K. Consequently, the pursuit of segmentation into seven clusters is not warranted.

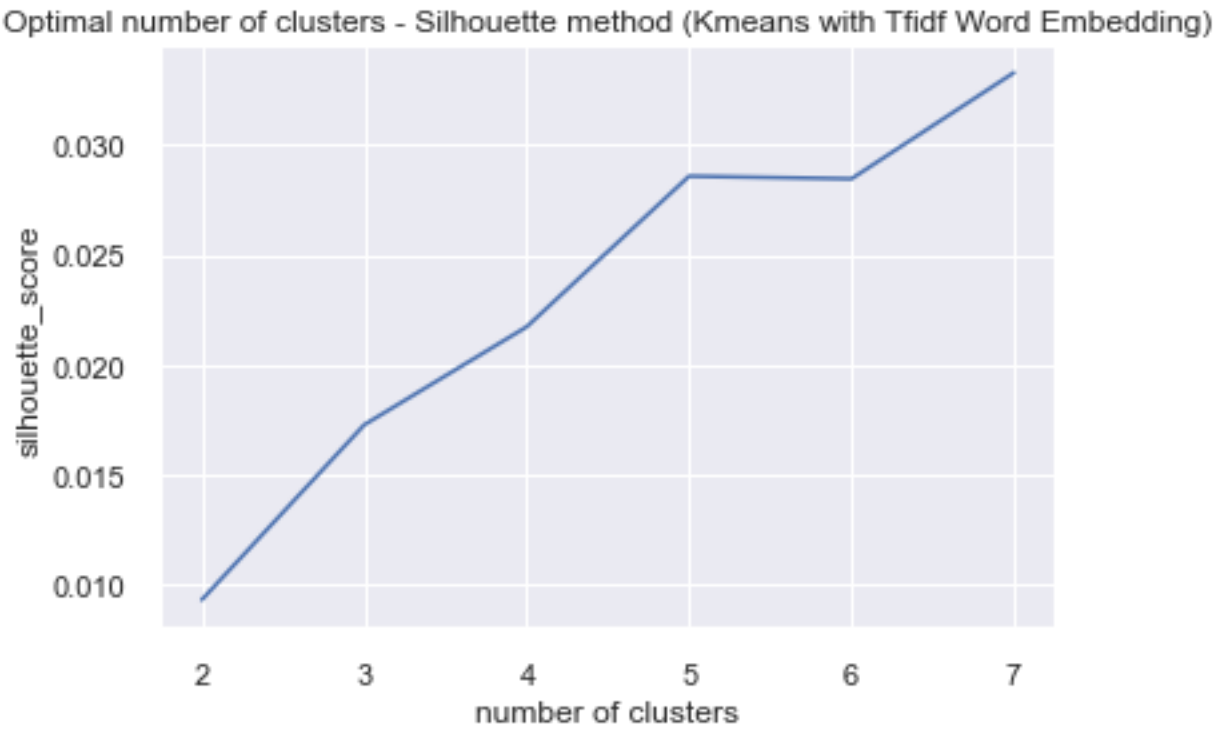

Figure 4.41: Optimal number of clusters - Silhouette method (K-means with TFIDF Embedding).

Within Figure 4.42, the visual representation elucidates the optimal determination of cluster quantity through the Silhouette method. Notably, the attainment of the maximal silhouette score at K = 7 indicates the optimal partitioning into seven clusters when employing the Mini-Batch K-means clustering approach with a TF-IDF embedding. Consequently, the decision to pursue segmentation into seven clusters is justified. As previously expounded, a silhouette score of one suggests that each data point is unlikely to be assigned to another cluster, while a score approximating zero indicates the ease with which each data point could be reassigned to an alternative cluster. Examination of Figure 4.42 reveals a silhouette score in close proximity to zero, implying the potential ease of reassignment of each data point to another cluster. It is crucial to exercise prudence in unequivocally relying on silhouette scores, especially when manifesting low values, as they may not consistently discern the optimal K. Therefore, the decision to pursue segmentation into seven clusters is not necessary.

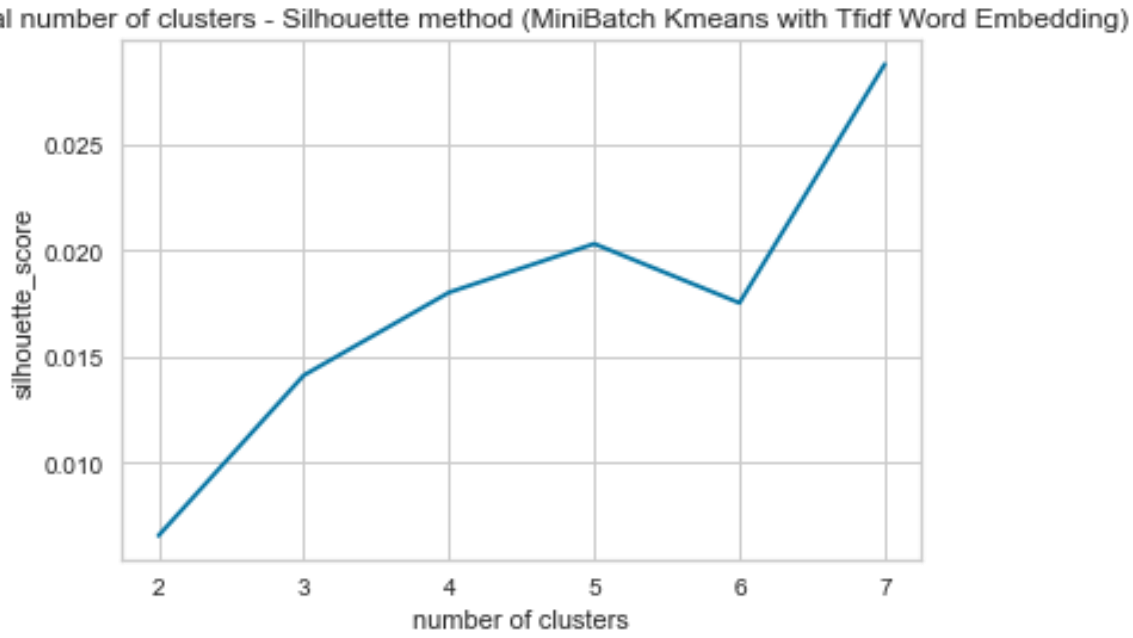

Figure 4.42: Optimal number of clusters - Silhouette method (Mini-Batch K-means with TFIDF Embedding).

It can be observed that Cluster 0, which contains authentic tweets related to COVID-19, has been assigned the label 0, whereas Cluster 1, comprising fabricated tweets about COVID-19, has been assigned the label 1. According to Table 4.14, the K-means algorithm assigned 13,093 tweets to Cluster 0 and 486,907 tweets to Cluster 1. Similarly, the mini-batch K-means algorithm assigned 38,114 tweets to Cluster 0 and 461,886 tweets to Cluster 1, based on the data provided.

Table 4.14: Number of observations per cluster (TF-IDF word embedding).

Model	Number of observations By Cluster
K-means	{1: 486907, 0: 13093}
mini-batch k-means	{1: 461886, 0: 38114}

Examining the areas of consensus between the two algorithms in their categorization could prove valuable. To achieve this, a two-dimensional matrix can be created, with one dimension representing the K-means clustering of tweets and the other dimension representing the mini-batch K-means clustering.

A confusion matrix would be an appropriate metric to utilize in this endeavour. Table 4.15 displays the crosstabulations generated by the mini-batch K-means and K-means algorithms using Bert word embedding.

Table 4.15: Crosstabulations (TF-IDF word embedding).

		K-means	
	Clusters	**0**	**1**
mini-batch k-means	**0**	0	38 114
	1	13 093	448 793

In Table 4.15, a 2x2 matrix is presented, and the sum of the diagonal elements represents the count of items that both clustering methods classified identically. These values account for 89.8% [(0+ 448 793)/500000] of all tweets, indicating that the two models agreed on the classification of this portion of the dataset. This proportion is the highest among the three different word embeddings discussed so far. In Section 4.2, we examined the labelled data, and Table 4.16 shows a 2x2 confusion matrix that illustrates our model's performance on this data. The summation of the diagonal elements within the matrix represents the count of accurately classified instances, while the rest of the values indicate the number of misclassified instances. Using the TF-IDF word embedding, the K-means model achieved an accuracy of 65% on the labelled data, while the Mini-Batch K-means model achieved a higher accuracy of 69%. It is worth noting that the Mini-Batch K-means algorithm significantly outperformed the K-means algorithm in terms of accuracy.

Table 4.16: Confusion Matrix Test Data (TF-IDF word embedding).

K-means							
		Actual				Actual	
	Tdidf	fake	real		Tdidf	fake	real
Predicted	fake	2,132	1,331	Predicted	fake	33%	21%
	real	928	2,029		real	14%	32%
Mini-Batch K-means							
		Actual				Actual	
	Tdidf	fake	real		Tdidf	fake	real
Predicted	fake	2,134	1,060	Predicted	fake	33%	17%
	real	926	2,300		real	14%	36%

The 2x2 crosstabulation matrix serves as a concise and informative representation of the frequency or count distribution of observations across different categories of the variables under study. Table 4.17 presents an extensive 2x2 crosstabulation matrix specifically relevant to the labelled data discussed in section 4.2. The cumulative sum of the diagonal elements within the matrix represents the items that both clustering methods have consistently assigned to the same class. This cumulative sum amounts to 95.2% ((2,937 + 3,174)/6,420) of all the tweets where both models have reached a consensus on the classification. This high level of agreement provides substantial evidence for the reliability and consistency of the clustering techniques in terms of their classification decisions. It is worth emphasizing that this finding indicates a notable degree of agreement between the two clustering methods. A closer examination of the off-diagonal elements within the matrix, which correspond to the instances where discrepancies emerged in the classifications between the clustering methods, reveals a relatively low level of divergence or uncertainty between the models. These findings underscore the robustness and reliability of the clustering techniques, as they demonstrate a consistent alignment in their classification outcomes for most of the observed tweets.

Table 4.17: Crosstabulation for the labelled data (TF-IDF word embedding).

		K-means	
	Clusters	**0**	**1**
mini-batch k-means	**0**	2,937	289
	1	20	3,174

Table 4.18 provides an illustration of the first 20 pre-processed tweets, where "Clean_Text" represents the pre-processed tweets discussed in section 4.1, "cluster" represents the clustering performed by the K-means algorithm, and "cluster2" denotes the clustering performed by the mini-batch K-means algorithm. It can be observed that the eighth tweet was assigned to cluster 1 by the K-means algorithm, whereas the same tweet was placed into cluster 0 by the mini-batch K-means algorithm.

Table 4.18: Example of the tweets that have been clustered by K-means and mini-batch K-means using TF-IDF word embedding.

Clean_Text	cluster	cluster2
brit reed caught kitten hospital confirmed cov...	1	1
impressive yes missing context question neurol...	1	1
going wager natural immunity letter rip pandem...	1	1
thanks castle theyre steep killing gamecovid y...	1	1
jayghai rio state immunization team excellent ...	1	1
plan continue wearing mask public mask mandate...	1	1
akeprofii day surveillance death specifically ...	1	1
dont remember bad player ducking smoke comment...	1	0
community health need helping stop spread covi...	1	1
didnt know beat covid uncertainty experience t...	1	1
happy covid shelter place day smoking collabor...	1	1
desi hirley oanguy aulril avid reeves damg uke...	1	1
health department ' fast loose covid profit ro...	1	1
covid increasing simple protect try stay home ...	1	1
administration confirmed making second covid b...	1	1
thanks partner positive met literally read mak...	1	1
accidental special checkers – evidence linking...	1	1
getting easier simply visit nearest vaccinatio...	1	1
important covid study infectious natural immu...	1	1
pacific islander news june audio language comm...	1	1

An analysis was conducted to identify the predominant words occurring within each cluster to determine the common words used in both genuine and fake tweets related to COVID-19. In order to attain a more profound comprehension of the language patterns in authentic and fabricated tweets related to COVID-19, we can analyse the most frequent words used in each cluster. Let us begin by examining cluster 0 of the K-means algorithm, which comprises 13,093 tweets (as shown in Table 4.14). By reviewing Figures 4.43 and 4.44, we can identify some of the most used words in this cluster, such as "justice," "dead," "company," "ministry," and more.

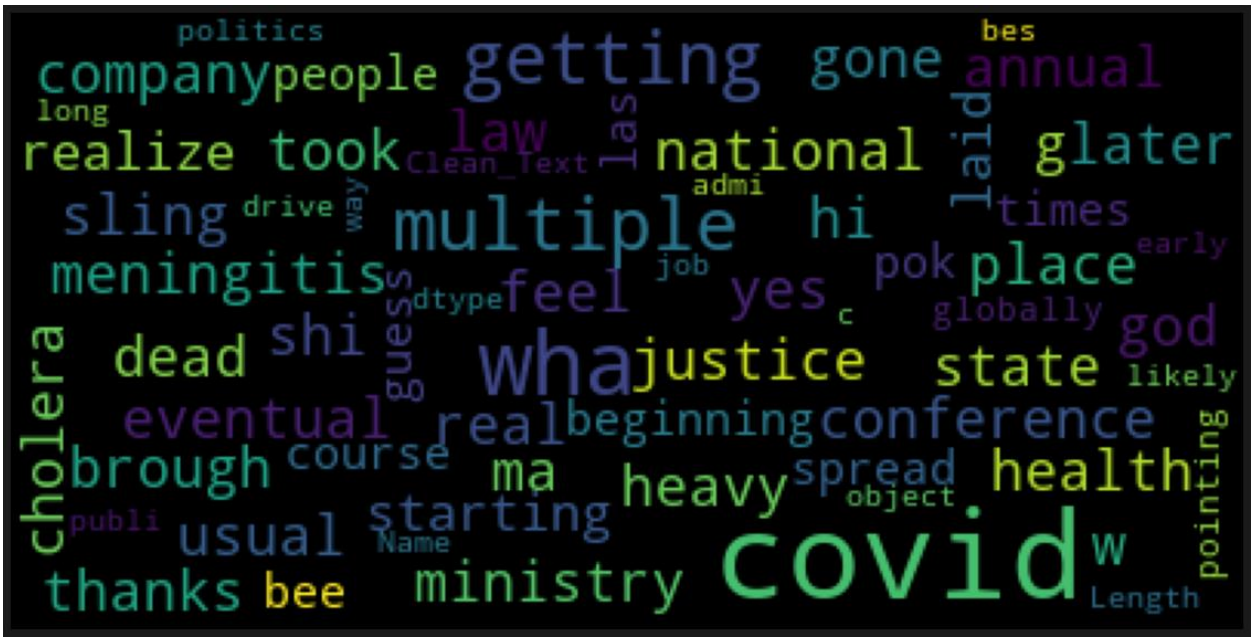

Figure 4.43 Word cloud for COVID-19 Tweets from cluster number 0 (real) from the K-means algorithm.

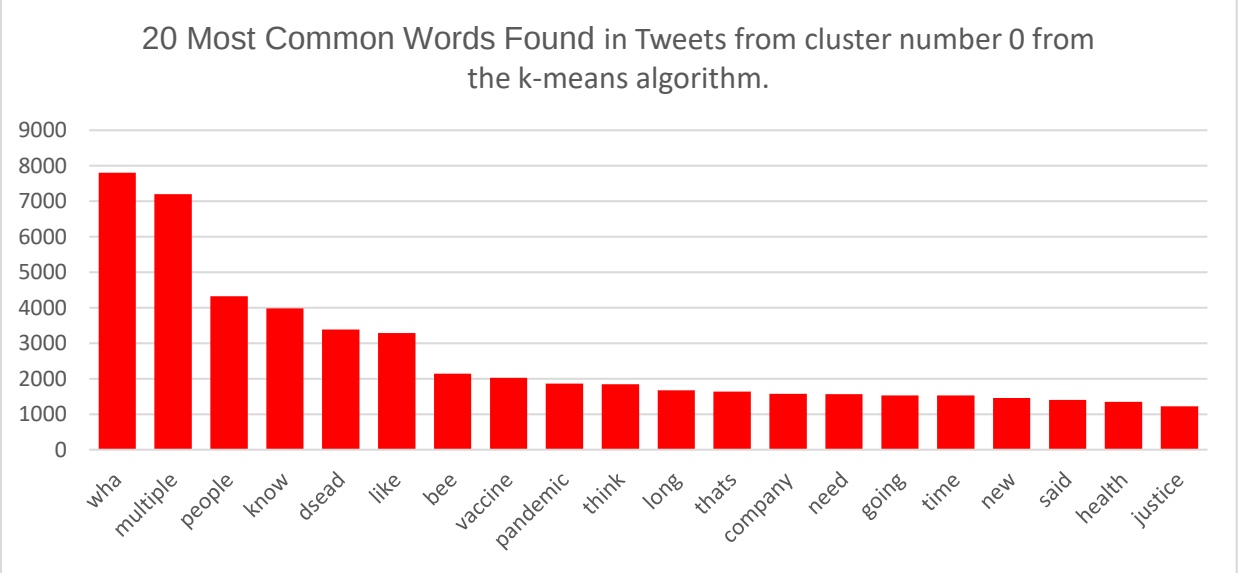

Figure 4.44 20 Most Common Words Found in Tweets from cluster number 0 (real) from the K-means algorithm.

Moving on to cluster number 1 of the K-means algorithm, which comprises 486,907 tweets as indicated in Table 4.14, we can extract the commonly used words. Some of the frequently occurring words in this cluster include new, vaccine, analytics, etc., as observed in Figures 4.45 and 4.46.

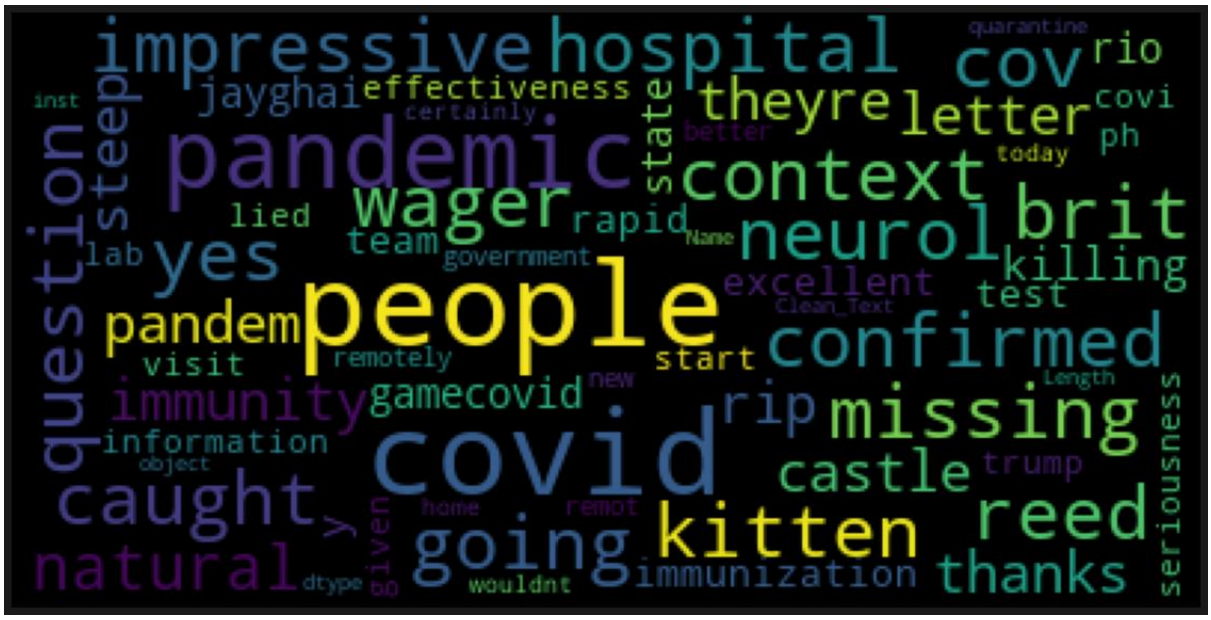

Figure 4.45 Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the K-means algorithm.

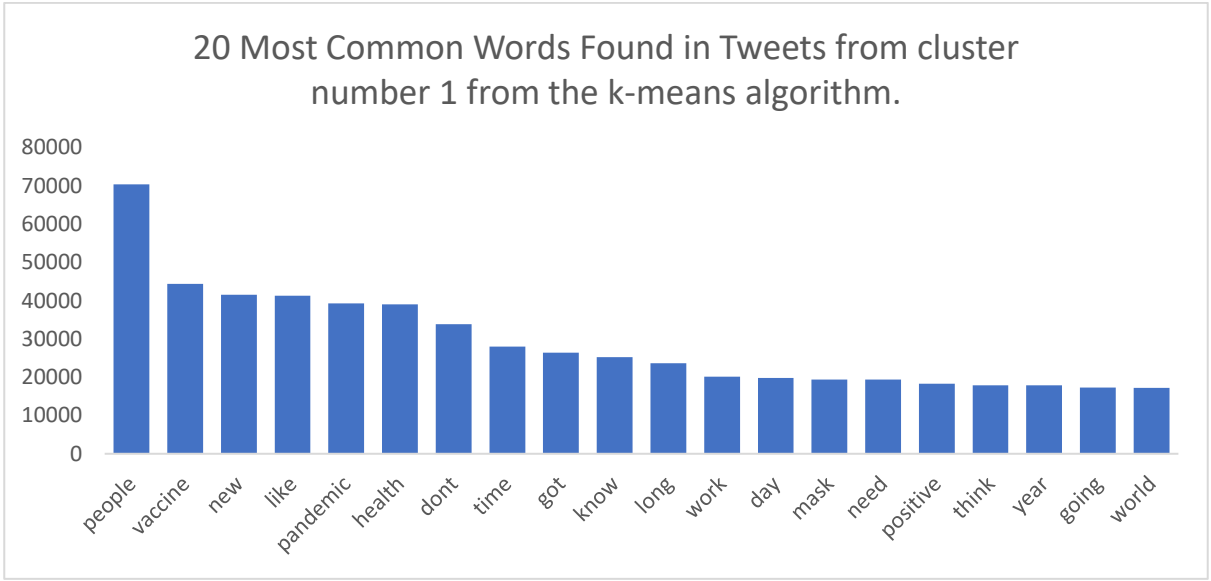

Figure 4.46 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the K-means algorithm.

To continue our analysis of the language used in genuine and fake tweets about COVID-19, we will now focus on cluster number 0 of the mini-batch K-means algorithm, which comprises 38,114 tweets as mentioned in Table 4.14.

Similar to the procedure employed for the K-means algorithm, the identification of the most used words in this cluster by examining Figures 4.47 and 4.48. These prominent words include "don't", "people", and "know".

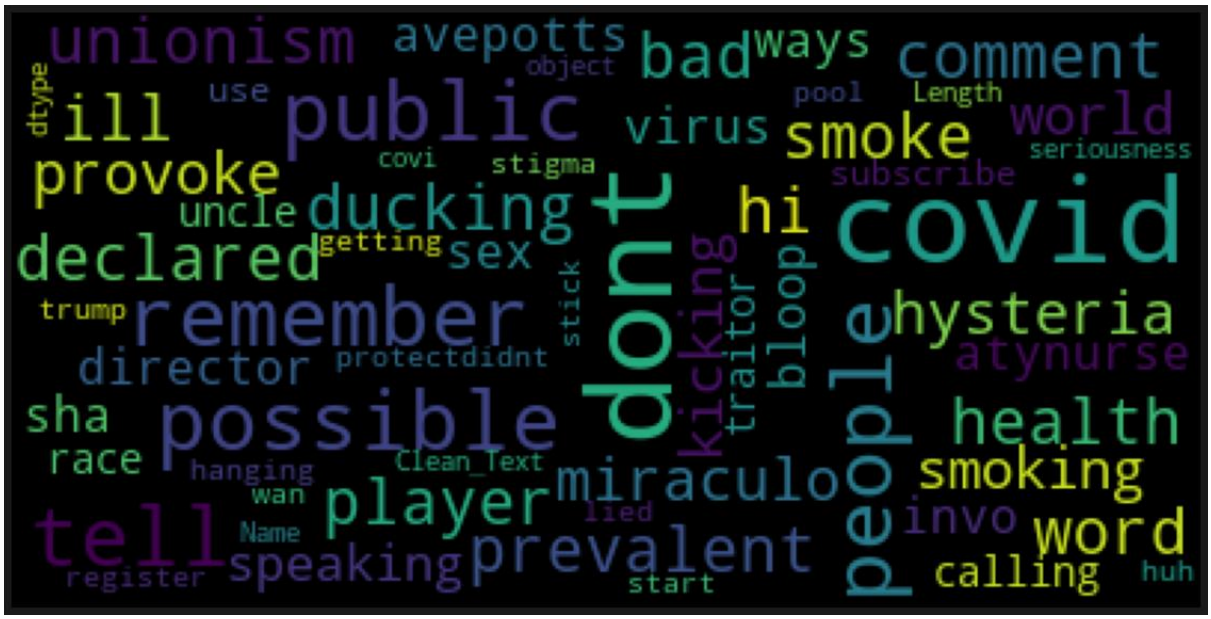

Figure 4.47 Word cloud for COVID-19 Tweets from cluster number 0 (real) from the mini-batch K-means algorithm

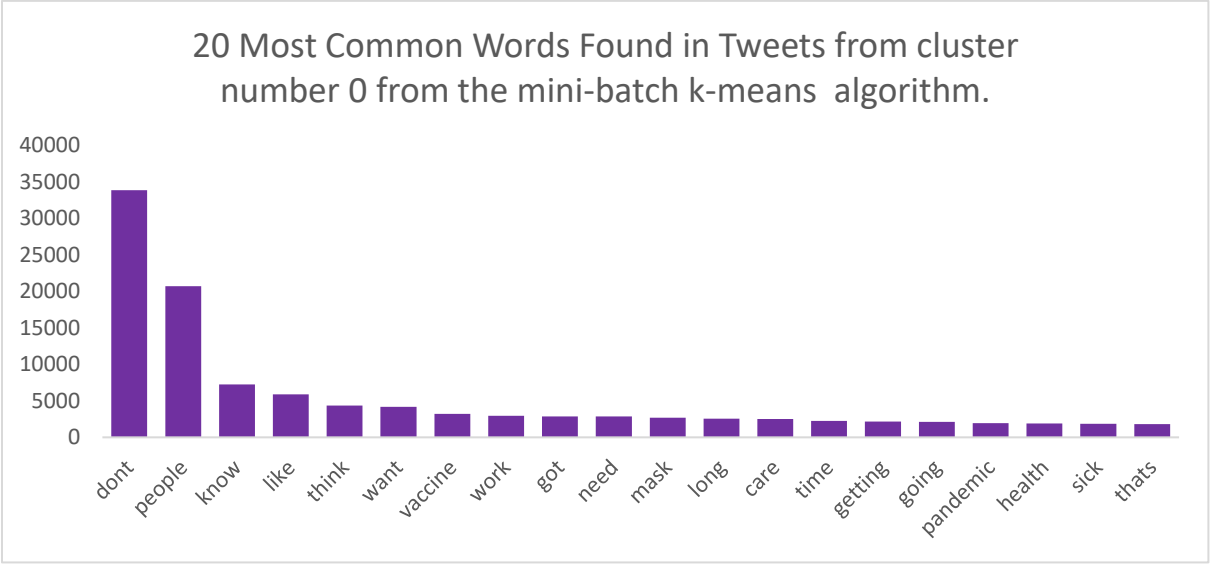

Figure 4.48 20 Most Common Words Found in Tweets from cluster number 0 (real) from the mini-batch K-means algorithm.

To complete our analysis of the language used in genuine and fake tweets about COVID-19, we will now examine cluster number 1 generated by the mini-batch K-means algorithm. This cluster comprises a total of 461,886 tweets, as mentioned in Table 4.14. By examining the most frequently occurring words in this cluster, such as "people," "vaccine," "new," "pandemic," and more, as depicted in Figure 4.49 and Figure 4.50, we can gain further insights into the topics and themes commonly associated with fabricated tweets about COVID-19.

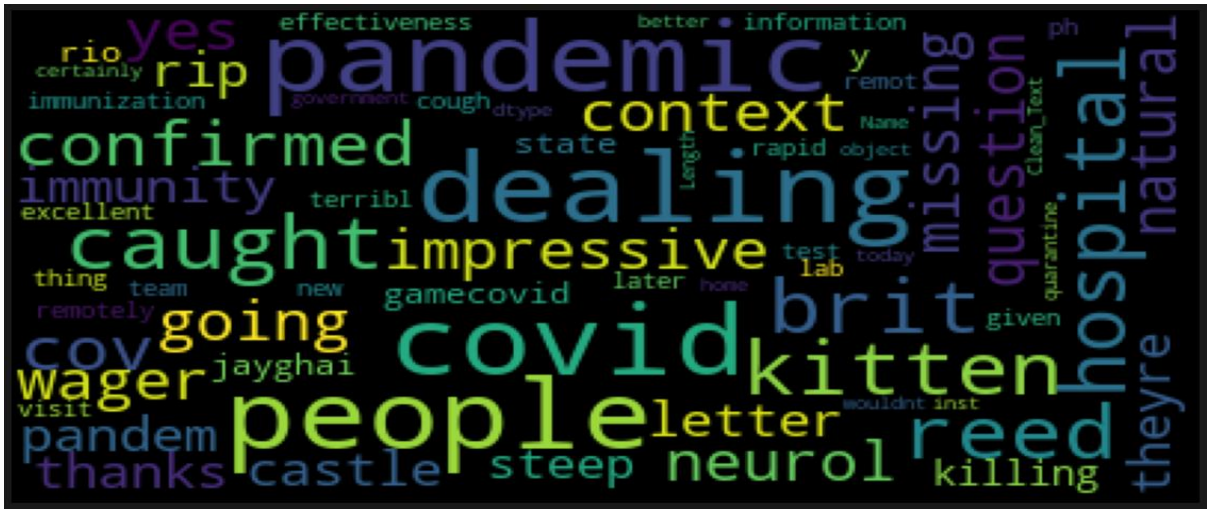

Figure 4.49 Word cloud for COVID-19 Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

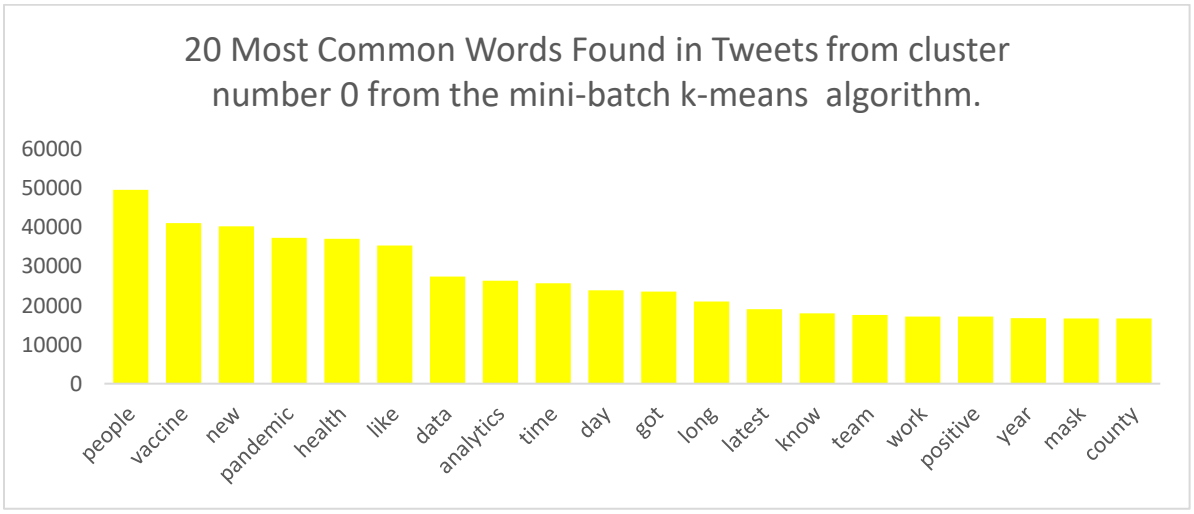

Figure 4.50 20 Most Common Words Found in Tweets from cluster number 1 (fake) from the mini-batch K-means algorithm.

To visualize distinct clusters, we performed dimensionality reduction using K-means algorithm-generated cluster labels. We selected a cluster count of 2 to analyse clusters as either fake or real COVID-19 tweets. We used t-SNE and Truncated SVD for dimensionality reduction. Figure 4.51 shows a two-dimensional representation of the clusters obtained through K-means algorithm. The left side displays a visualization of t-SNE dimensionality reduction method's representation of the clusters, while the right side shows visualization using the Truncated SVD dimensionality reduction method. In the sub-figure on the right, cluster 1 has a triangular shape, while the sub-figure on the left displays a more rounded shape. The blue-coloured cluster represents cluster 0, while the orange-coloured cluster represents cluster 1. Both t-SNE and Truncated SVD effectively visualized the clusters. The right figure displays the two clusters, cluster 0 and cluster 1, clearly.

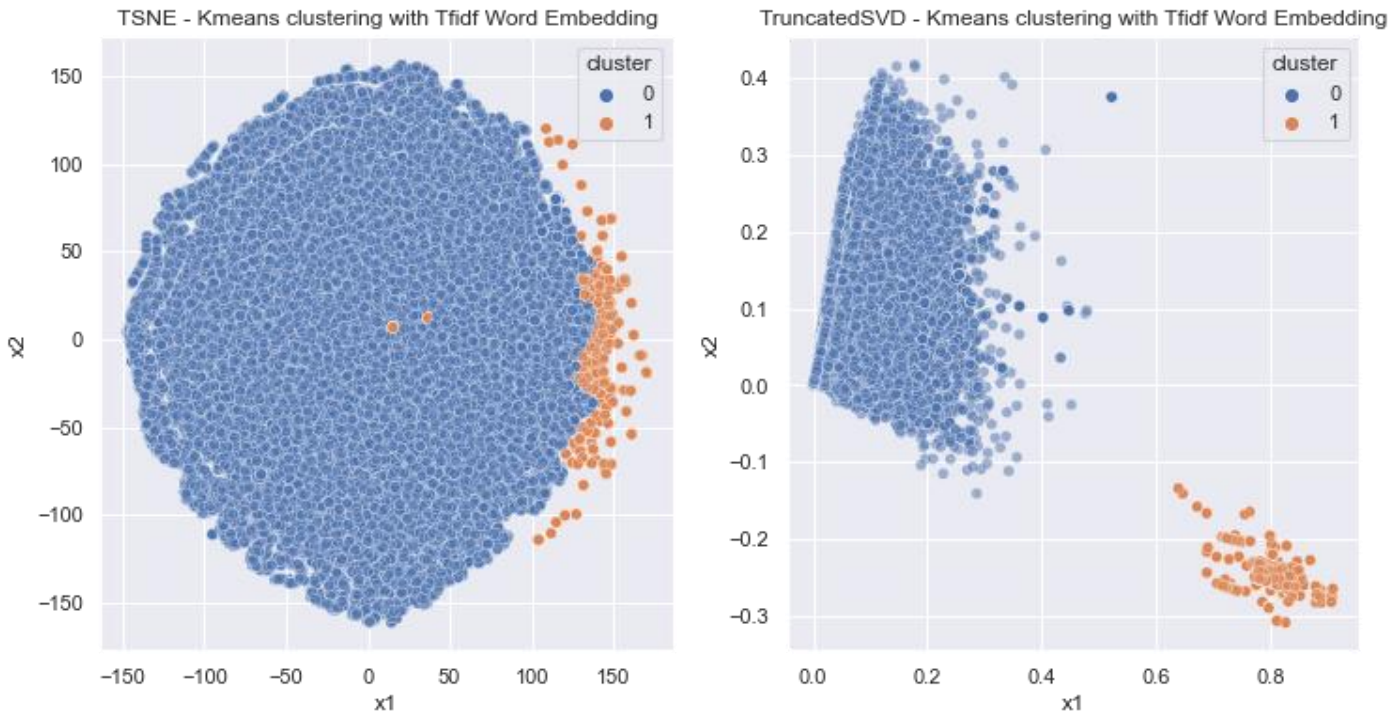

Figure 4.51: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) - K-means with TF-IDF Word Embedding.

Figure 4.52 displays the clusters obtained by applying the Mini Batch K-means algorithm to t-SNE and Truncated SVD reduced dimensions. The left-hand side of the figure represents the t-SNE dimensionality reduction method, while the right-hand side shows the Truncated SVD dimensionality reduction method. In the sub-figure on the right, cluster 1 has a triangular shape, whereas the sub-figure on the left shows a more rounded shape for the same cluster. The blue coloured cluster corresponds to cluster 0, while the orange coloured cluster represents cluster 1. While the visualizations generated by t-SNE and Truncated SVD appear to show three clusters, we can confidently conclude that there are only two clusters as indicated by the two colours. These tweets could potentially be outliers in the data, possibly containing some degree of truth.

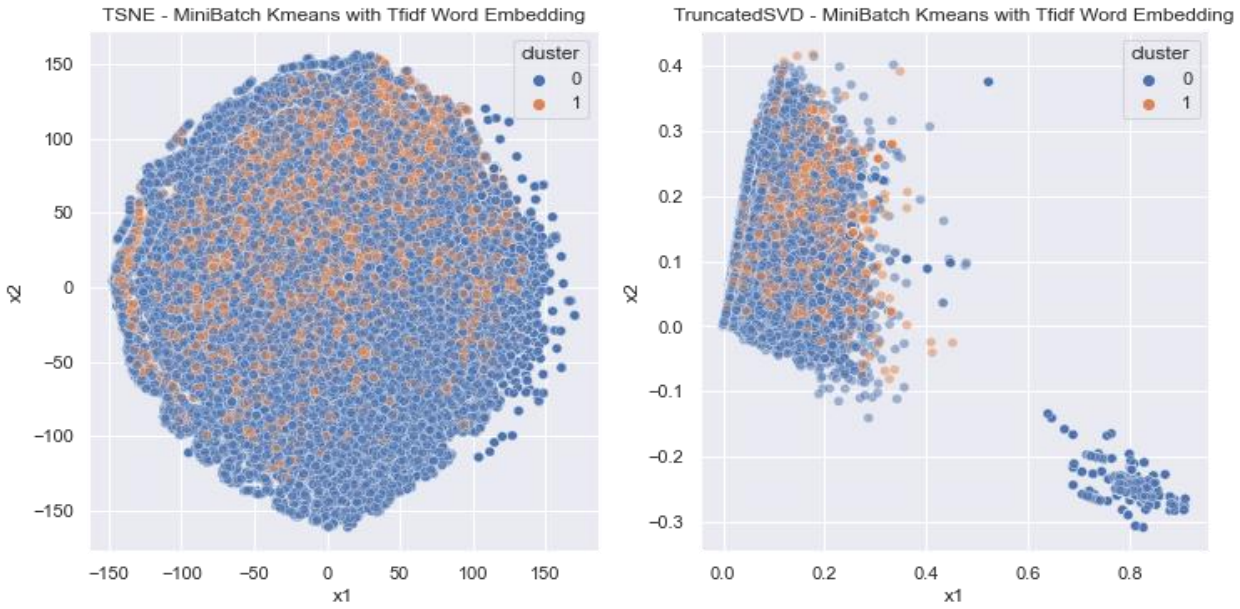

Figure 4.52: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) – Mini Batch K-means with TF-IDF Word Embedding.

We present visualizations of the clusters identified in the labelled data discussed in section 4.2. Figure 4.53 provides comprehensive two-dimensional representations of the clusters obtained through the utilization of the K-means algorithm with TF-IDF word embedding. The left figure showcases the visualization based on the t-SNE dimensionality reduction method, while the right figure depicts the visualization derived from the Truncated SVD dimensionality reduction technique.

Within these visualizations, the blue cluster corresponds to cluster 0, while the orange cluster represents cluster 1. Notably, the cluster portrayed in the right sub-figure exhibits a discernible triangular-shaped pattern, while the left sub-figure displays a more intricate web-like structure.

Both the t-SNE and Truncated SVD techniques effectively capture and encapsulate the unique characteristics of the clusters within the two-dimensional space. These visualizations serve a crucial purpose in enabling clear differentiation between cluster 0 and cluster 1, thereby facilitating the straightforward identification of the two clusters. Through the visual representations, it becomes evident how the clusters are distinct and separate from one another.

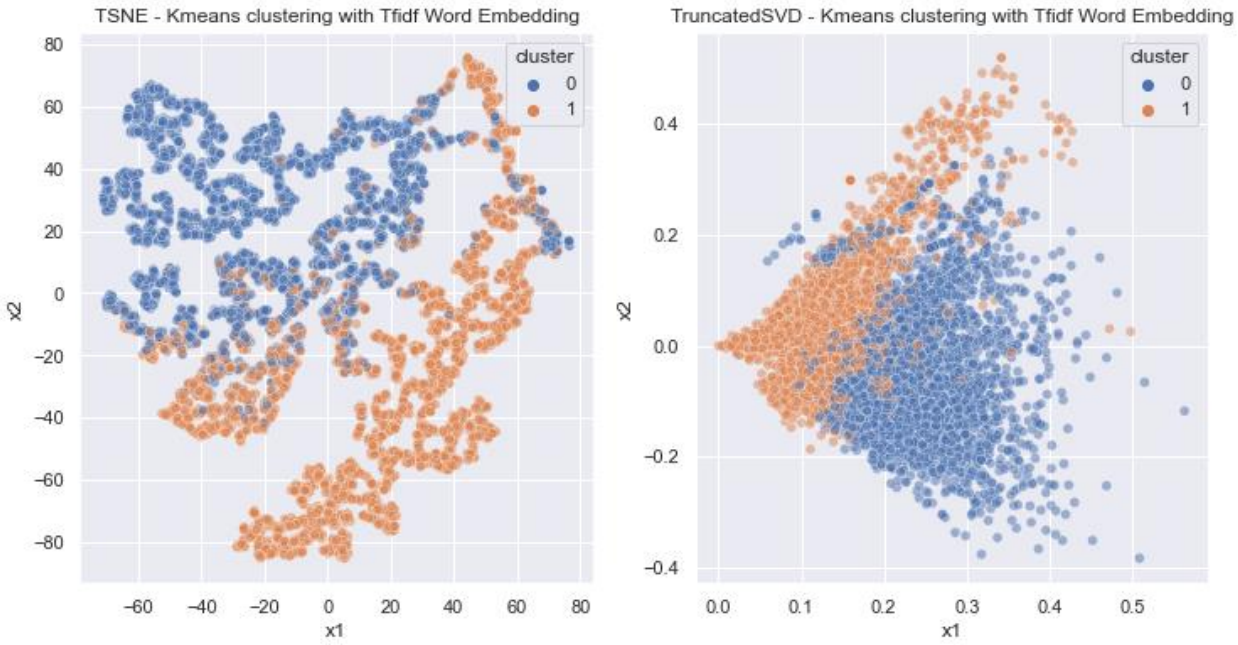

Figure 4.53: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization – K-means with TF-IDF Word Embedding (labelled data).

Figure 4.54 illustrates a graphical representation of the clusters formed through the application of the Mini-Batch K-means algorithm with TF-IDF word embedding to the labelled data discussed in section 4.2. The left figure demonstrates the utilization of the t-SNE dimensionality reduction technique to represent the clusters, while the right figure utilizes the Truncated SVD dimensionality reduction method for visualization.

The clusters are denoted as Cluster 0 and Cluster 1, portrayed in blue and orange colours, respectively. Upon meticulous scrutiny of the visualizations, it becomes apparent that the cluster portrayed in the right sub-figure exhibits a conspicuous triangular-shaped pattern, whereas the left sub-figure displays a more intricate web-like structure. However, it is noteworthy that both the t-SNE and Truncated SVD techniques employed in this instance were not as proficient in accurately capturing the distinct cluster as demonstrated in Figure 4.53 and Figure 4.54.

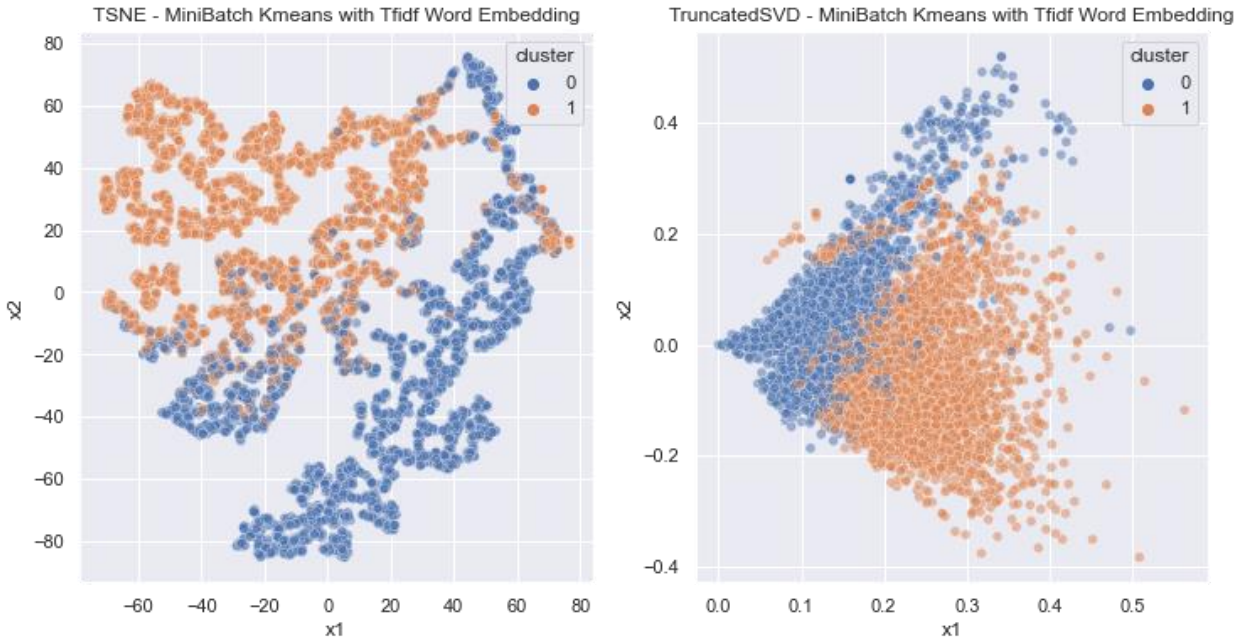

Figure 4.54: t-SNE (Left) & Truncated SVD dimensionality reduction (Right) visualization - Mini Batch K-means with TF-IDF Word Embedding (labelled data).

4.4 DISCUSSION

Whenever we use Machine Learning, it is essential to come up with a way to measure the algorithm's performance. The confusion matrix is a valuable performance measure for classification tasks, which provides practitioners with a visual insight into how their algorithm is performing. The confusion matrix can provide various metrics that can assist in measuring the performance of our classification models. Accuracy, Precision, Recall, and F1 Score are just a few metrics that we have looked at in this section.

Accuracy has already been discussed in this paper; we will now delve into the other metrics.

- Precision informs us of the amount of actual positive labels from all the labels our classifier has labelled as positive.

  Precision = $\frac{True\ Positives\ (TP)}{True\ Positives\ (TP)+False\ Positives(FP)}$

- Recall informs us of the number of positive labels that our classifier correctly labelled as positive.

  Recall = $\frac{True\ Positives\ (TP)}{True\ Positives\ (TP)+False\ Negatives\ (FN)}$

- The F1 Score constitutes a composite metric that incorporates both Precision and Recall, providing a consolidated measure of their performance. It reaches its maximum value when Precision is equal to Recall. In practical scenarios, as efforts are made to enhance model precision, the recall tends to decrease, and vice versa. The F1 score effectively captures both trends within a single value. A higher F1 score indicates a superior model performance.

  F1 Score = 2*$\frac{Precision*Recall}{Precision+Recall}$

In the context of using word2vec word embeddings, K-means model and Mini-Batch K-means model can be applied to cluster similar words or sentences based on their vector representations. When using word2vec embeddings with K-means, the model demonstrated an accuracy of 60.1%, precision of 58.9%, recall of 53.6%, and F1 score of 56.1%. This indicates that the model can cluster similar words together with a moderate level of accuracy. On the other hand, the Mini-Batch K-means algorithm represents a variation of the traditional K-means approach, wherein it utilizes a random subset of the data during each iteration, making it more efficient for large datasets. When using word2vec embeddings with mini-batch K-means, the model demonstrated an accuracy of 39.9%, precision of 40.7%, recall of 57.0%, and F1 score of 47.5%. This observation signifies that the Mini-Batch K-means model exhibits comparatively inferior accuracy when contrasted with the standard K-means model, manifesting reduced precision and F1 score. However, it has a higher recall, suggesting that it is better at identifying similar words that should be grouped together.

These results suggest that the K-means model is better at clustering tweets to either fake or real tweets about COVID-19 compared to the Mini-Batch K-means model. This could be because the K-means model clusters the data points using the entire dataset in each iteration, which can lead to a more accurate representation of the data points. Alternatively, the Mini-Batch K-means model only uses a subset of the data in each iteration, which can lead to a less accurate clustering. The Mini-Batch K-means algorithm correctly predicted that around 46% of the identified positive tweets were actually positive. BERT, an advanced transformer-based language model, can generate contextualized embeddings for words or sentences by taking into account their surrounding context within a larger sentence or document. This state-of-the-art approach enhances the representation of linguistic elements, thereby facilitating more accurate and nuanced language understanding tasks, (Devlin et al, 2019). In the context of using Bert word embeddings, K-means model and Mini-Batch K-means model was used to cluster tweets to either fake or real tweets about COVID-19. The K-means model demonstrated a performance with an accuracy of 75.9%, precision of 82.8%, recall of 62.3%, and F1 score of 71.1%. This suggests that the model can effectively cluster words or sentences with a high level of similarity. The Mini-Batch K-means model demonstrated an accuracy of 47.4%, precision of 45.7%, recall of 54.6%, and F1 score of 49.7%. This indicates that the Mini-Batch K-means model is less effective in clustering tweets to either fake or real about COVID-19 compared to the K-means model. Overall, the K-means model using Bert word embeddings outperforms the Mini-Batch K-means model using Bert word embeddings, suggesting that K-means clustering with Bert word embeddings can be a powerful tool for natural language processing tasks. The high accuracy, precision, and F1 score of the K-means model using Bert embeddings indicate that it can be used to effectively cluster tweets to either fake or real tweets about COVID-19 and extract useful insights from similar text data. It can be inferred from the recall that roughly 54% of the positive classes were predicted accurately. TF-IDF, a widely adopted technique, is employed to transform words or sentences into vectors by considering their frequency of occurrence within a document or corpus.

In the context of using TF-IDF word embeddings, the K-means model and Mini-Batch K-means model was used to cluster tweets to either fake or real tweets about COVID-19 based on their TF-IDF vector representations. The K-means model using TF-IDF word embeddings has an accuracy of 64.8%, precision of 61.6%, recall of 69.7% and F1 score of 65.4%. This suggests that the model can effectively cluster words or sentences with a high level of similarity. Alternatively, the Mini-Batch K-means model using TF-IDF word embeddings has an accuracy of 60.1%, precision of 58.9%, recall of 53.6% and F1 score of 56.1%. This indicates that the Mini-Batch K-means model is less effective in clustering was used to cluster tweets to either fake or real tweets about COVID-19 compared to the K-means model. The K-means model using TF-IDF word embeddings appears to perform better than the Mini-Batch K-means model in this scenario, suggesting that K-means clustering with TF-IDF word embeddings can be a useful in clustering similar text data.
According to the metrics in table 19, it is also noticed that K-means clustering using Bert is best performing model for the test data used. As the result shown, K-means clustering using Bert achieved highest accuracy (75.9%), achieved highest precision (82.8%), K-means clustering using TF-IDF achieved highest recall (69.7%), and K-means clustering using Bert achieved highest F1 (71.1%). Table 4.19 shows the model performance comparison with different metrics.

Table 4.19: Model performance comparison with different metrics.

Model	Accuracy	Precision	Recall	F1
K-means clustering using Word2Vec	**60.1%**	**58.9%**	**53.6%**	**56.1%**
Mini Batch K-means clustering using Word2Vec	**39.9%**	**40.7%**	**57.0%**	**47.5%**
K-means clustering using Bert	**75.9%**	**82.8%**	**62.3%**	**71.1%**
Mini Batch K-means clustering using Bert	**47.4%**	**45.7%**	**54.6%**	**49.7%**
K-means clustering using TF-IDF	**64.8%**	**61.6%**	**69.7%**	**65.4%**
Mini Batch K-means clustering using TF-IDF	**69.1%**	**66.8%**	**69.7%**	**68.2%**

The next chapter will centre on the study's conclusions.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The final chapter of this thesis will address the problem statement posed in section 1.3, drawing on the findings from the previous chapter. Additionally, we will explore potential future work that could prove beneficial. The present study utilized unsupervised algorithms to identify fake news (fake tweets concerning COVID-19). This task is substantially more challenging with unsupervised algorithms than with their supervised counterparts. The lack of labelled data presents a significant hurdle in modelling the distinction between genuine and fake tweets. In order to detect fake tweets pertaining to COVID-19, we employed both the Mini-Batch K-means and K-means algorithms, incorporating three distinct word embeddings as part of our methodology. We performed multiple experiments on the test dataset to evaluate the accuracy of all six models. The outcomes indicate that the K-means algorithm outperformed other algorithms in detecting fake tweets about COVID-19. Table 5.1 illustrates that the K-means algorithm utilizing Bert word embedding was the most effective model, and the Mini-Batch K-means algorithm using TF-IDF word embedding was the second-best performing model.

Our study demonstrated that all four models yielded satisfactory outcomes, except for two models which were Mini-Batch K-means with Word2Vec and Bert word embedding that performed poorly. When considering all three different word embeddings, K-means had an overall accuracy of 67%, whereas the Mini-Batch K-means algorithm only attained 52%.

Table 5.1: Summary accuracy results from the 6 models.

Accuracy	Word2Vec	TF-IDF	Bert	Average Row
K-means	60%	65%	76%	**67%**
Mini-Batch K-means	40%	69%	47%	**52%**
Average columns	**50%**	**67%**	**62%**	

5.2 FUTURE WORK

Due to time constraints during the paper's development, a comprehensive exploration of all potential avenues and the complete development of emergent ideas could not be undertaken. There are several important questions that require further attention, namely:

- Exploring the use of other clustering algorithms like hierarchical clustering and DBSCAN (DBSCAN does not behave nicely when sample sizes are very large, (Maharjan, 2018).
- Test more metrics, like how much time (runtime) it takes to run any of the model, to check at what cost does accuracy come at, in terms of time.
- Measure which model uses less computational resource, to check at what cost does accuracy come at, in terms of computational resource.
- Explore other word embedding methods, to check what impact that has on accuracy.

Future studies can focus on any of the above.

REFERENCES:

Adali S. and Horne B.D (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news [online]. Available from: https://arxiv.org/pdf/1703.09398.pdf (accessed 29 May 2022).

Alaimo C. and Kallinikos J (2017). Social Media and Fake News in the 2016 Election [online]. Available from: https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.31.2.211 (accessed 23 January 2022).

Allcott H. and Gentzkow M. (2016). Social Media and Fake News in the 2016 Election [online]. Available at: https://www.jstor.org/stable/pdf/44235006.pdf?refreqid=excelsior%3Aa0dfb47eba389194d4e1532c34f53a19&ab_segments=&origin=&initiator=&acceptTC=1 (accessed 7 January 2022).

Akdogan A. (2021). Word Embedding Techniques: Word2Vec and TF-IDF Explained [online]. Available at: https://towardsdatascience.com/word-embedding-techniques-word2vec-and-tf-idf-explained-c5d02e34d08 (accessed 5 April 2022).

Allcott H. and Gentzkow M. (2017). Social media and fake news in the 2016 election [online]. Available at: https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.31.2.211 (accessed 9 February 2022).

Agarwal N. (2022). The Ultimate Guide To Different Word Embedding Techniques In NLP [online]. Available at: https://www.kdnuggets.com/2021/11/guide-word-embedding-techniques-nlp.html) (accessed 27 November 2022).

Andersen K.G., Rambaut A., Ian Lipkin W., Holmes E.C. and Garry R.F. (2020). The proximal origin of SARS-CoV-2 [online]. Available at: https://www.nature.com/articles/s41591-020-0820-9 (accessed 6 February 2022).

Ahmed H., Traore I. and Saad S. (2017), Detecting opinion spams and fake news using text classification [online]. Available from: https://www.researchgate.net/publication/322128415_Detecting_opinion_spams_and_fake_news_using_text_classification(accessed 24 February 2022).

Ahmad I., Yousaf M., Yousaf S. and Ahmad M.O. (2020). Fake News Detection Using Machine Learning Ensemble Methods [online]. Available at: https://www.hindawi.com/journals/complexity/2020/8885861/ (accessed 20 January 2022).

Asha J. and Meenakowshalya A. (2021), Fake News Detection Using Unsupervised and Deep Learning Algorithms [online]. Available from: https://www.researchgate.net/publication/358723053_Fake_News_Detection_Using_Unsupervised_and_Deep_Learning_Algorithms (accessed 07 November 2023).

Banerji A. (2023), K-Mean: Getting the Optimal Number of Clusters, [online]. Available from: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/ (accessed 17 November 2022).

Battaglia M. (2013). Convenience Sampling [online]. Available at: http://srmo.sagepub.com/view/encyclopedia-of-survey-research-methods/n105.xm (accessed 13 March 2022).

BBC News (2020). The people who think coronavirus is caused by 5G [online]. Available from: https://www.bbc.com/news/av/stories-53285610 (accessed 30 January 2022).

Biral E. (2021).The impact of Epidemics on European society: a historical perspective of socio-economic change during pandemic periods [online]. Available from: http://dspace.unive.it/bitstream/handle/10579/20900/881499-1256578.pdf?sequence=2 (accessed 23 January 2022).

BusinessOfApps (2023), Twitter Revenue and Usage Statistics (2023) [online]. Available at: https://www.businessofapps.com/data/twitter-statistics/ (accessed 27 March 2022).

Chavan M.M., Patil A., Dalvi L. and Patil A. (2015). Mini Batch K-Means Clustering On Large Dataset [online].
Available at: http://ijsetr.com/uploads/462153IJSETR4282-245.pdf (accessed 27 April 2022).

Coldwell D. and Herbst F. (2004). *Business Research*. Cape Town: Juta.
Cohen M.J. (2020), Does the COVID-19 outbreak mark the onset of a sustainable consumption transition? [online]. Available from: https://www.tandfonline.com/doi/full/10.1080/15487733.2020.1740472 (accessed 1 February 2022).
Conroy N.J, Rubin V.L and Chen Y (2015). Automatic deception detection: Methods for finding fake news [online]. Available from: https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/pra2.2015.145052010082 (accessed 17 January 2022).
Constine J. (2020). Facebook Deletes Brazil President's Coronavirus Misinfo Post. Tech Crunch [online]. Available online at: https://techcrunch.com/2020/03/30/facebook-removes-bolsonaro-video/ (accessed March 31, 2022).
Dahouda M. and Joe I. (2021). A Deep-Learned Embedding Technique for Categorical Features Encoding [online]. Available at: https://www.researchgate.net/publication/353857384_A_Deep-Learned_Embedding_Technique_for_Categorical_Features_Encoding (accessed 27 June 2022).
Devlin J., Chang M. W., Lee K. and Toutanova K. (2019). BERT:Pretraining of deep bidirectional transformers for language understanding [online]. Available online at: https://aclanthology.org/N19-1423.pdf (accessed 25 April 2022).
Duffy A., Tandoc E. and Ling R. (2019), Too good to be true, too good not to share: the social utility of fake news [online]. Available online at: https://www.tandfonline.com/doi/full/10.1080/1369118X.2019.1623904 (accessed 05 March 2022).
Dusmanu M., Cabrio E. and Villata S. (2017). Argument Mining on Twitter: Arguments, Facts and Sources [online]. Available at: https://aclanthology.org/D17-1245/ (accessed 24 January 2022)
Ferrara A. and Montanelli S. (2017). Unsupervised Detection of Argumentative Units though Topic Modeling Techniques [online]. Available at: https://www.researchgate.net/publication/322587694_Unsupervised_Detection_of_Argumentative_Units_though_Topic_Modeling_Techniques (accessed 27 February 2022).

Forgy E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics, Vol. 21, pp. 768-780.
Fernández-Gavilanes M., Álvarez-López T., Juncal-Martínez J., Costa-Montenegro E. and González-Castaño F.J. (2016). Unsupervised method for sentiment analysis in online texts [online]. Available online at: https://www.sciencedirect.com/science/article/pii/S0957417416301300 (accessed 05 August 2022).
Gangireddy S. C., Padmanabhan D., Long C. and Chakraborty T. (2020), Unsupervised Fake News Detection: A Graph-based Approach [online]. Available online at: https://pureadmin.qub.ac.uk/ws/files/212663108/ht20_crc.pdf (accessed 05 February 2022).
Granik M. and Mesyura V. (2017). Fake news detection using naive Bayes classifier [online]. Available online at: https://ieeexplore.ieee.org/document/8100379/references#references (accessed 16 February 2022).
Hemalatha, I., Saradhi Varma G.P. and Govardhan, A. (2012), Preprocessing the Informal Text for efficient Sentiment Analysis [online]. Available online at: https://www.researchgate.net/profile/Indukuri-Latha/publication/334672058_Preprocessing_the_Informal_Text_for_efficient_Sentiment_Analysis/links/5d398f7c299bf1995b48cf41/Preprocessing-the-Informal-Text-for-efficient-Sentiment-Analysis.pdf (accessed 10 September 2022).
Hemalatha, I., Saradhi Varma G.P. and Govardhan, A. (2014). Case Study on Online Reviews Sentiment Analysis Using Machine Learning Algorithms [online]. Available online at: https://www.researchgate.net/publication/334672115_Case_Study_on_Online_Reviews_Sentiment_Analysis_Using_Machine_Learning_Algorithms (accessed 16 September 2022).
Hosseinimotlagh S. and Papalexakis E.E. (2016). Unsupervised Content-Based Identification of Fake NewsArticles with Tensor Decomposition Ensembles [online]. Available online at: https://www.researchgate.net/publication/323387293_Unsupervised_Content-

Based_Identification_of_Fake_News_Articles_with_Tensor_Decomposition_Ensembles (accessed 16 March 2022).
IBM (2020). What is natural language processing (NLP)? [online]. Available at: https://www.ibm.com/topics/natural-language-processing (accessed 7 May 2022).
Jia W., Sun M., Lian J. and Hou S. (2022). Feature dimensionality reduction: a review [online]. Available at:https://link.springer.com/article/10.1007/s40747-021-00637-x (accessed 10 April 2022).
Jockers M. (2020). Introduction to the Syuzhet Package [online]. Available at: https://perma.cc/9BN2-F3N3 (accessed 17 October 2022).
Kaufman L. and Rousseeuw P. (1990), An overview of online fake news: Characterization, detection, and discussion [online]. Available from: https://www.researchgate.net/publication/220695963_Finding_Groups_in_Data_An_Introduction_To_Cluster_Analysis (accessed 17 November 2022).
Karani D. (2020). Introduction to Word Embedding and Word2Vec [online]. Available at: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa (accessed 27 September 2022).
Kumar A.C. (2009). Analysis of unsupervised dimensionality reduction techniques [online]. Available at: https://www.doiserbia.nb.rs/Article.aspx?id=1820-02140902217K (accessed 17 April 2022).
Kumar F., Morstatter S. and Liu H. (2014). Twitter Data Analytics [online]. Available at: , https://www.researchgate.net/publication/316806338_Twitter_Data_Analytics (accessed 27 April 2022).
LIPPI M. and TORRONI P. (2016). Argumentation Mining: State of the Art and Emerging Trends [online]. Available at: https://www.researchgate.net/publication/299548392_Argumentation_Mining (accessed 10 February 2022).
Maharjan A. (2018). Nepali Document Clustering using K-Means, Mini-Batch K-Means, and DBSCAN [online]. Available at: https://elibrary.tucl.edu.np/bitstream/123456789/10020/1/thesis.pdf (accessed 10 March 2022).

Manning, C. D., Raghavan, P., and Sch¨utze, H. (2008). Introduction to information retrieval. Cambridge,England: Cambridge University Press.
Meehan K. (2020), Gender Classification using Twitter Text Data [online]. Available online at: https://www.researchgate.net/publication/344003069_Gender_Classification_using_Twitter_Text_Data (accessed 11 March 2022).
Mitchell T.M. (1997). Machine Learning [online]. Available online at: https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf (accessed 11 January 2022).
Mitchell A., and Oliphant J. B. (2020). Americans Immersed in COVID-19 News; Most Think Media Are Doing Fairly Well Covering It. Pew Research Center [online]. Available from: : https://www.journalism.org/2020/03/18/americans-immersed-in-covid-19-news-most-think-media-are-doing-fairly-well-covering-it/ (accessed 18 March 2022).
Ofcom (2020). Half of UK adults exposed to false claims about coronavirus [online]. Available at: https://www.ofcom.org.uk/news-centre/2020/half-of-uk-adults-exposed-to-false-claims-about-coronavirus (accessed 21 March 2022).
Oskolkov N. (2022), Dimensionality Reduction [online]. Available from: https://link.springer.com/chapter/10.1007/978-3-030-88389-8_9 (accessed 27 October 2022).
Pang B. and Lee L. (2008). Opinion Mining and Sentiment Analysis [online]. Available at: https://www.nowpublishers.com/article/Details/INR-011 (accessed 15 January 2022).
Park J., Blake C. and Cardie C. (2015). Toward machine-assisted participation in erulemaking: an argumentation model of evaluability [online]. Available at: https://doi.org/10.1145/2746090.2746118 (accessed 2 May 2022).
Peldszus A. and Stede M. (2013). From argument diagrams to argumentation mining in texts: A survey [online]. Available at: https://www.ling.uni-potsdam.de/~peldszus/ijcini2013-preprint.pdf (accessed 12 September 2022).
Pennycook G., Cannon T. and Rand, D. G. (2018). Prior exposure increases perceived accuracy of fake news [online].

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6279465/pdf/nihms972315.pdf (accessed 3 June 2022).

Pérez-Rosas V., B. Kleinberg B., Lefevre A. and Mihalcea R. (2017). Automatic Detection of Fake News [online]. Available at: https://www.researchgate.net/publication/319255985_Automatic_Detection_of_Fake_News (accessed 30 January 2022).

Pogiatzis A. (2019). NLP: Contextualized word embeddings from BERT [online]. Available at: https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b (accessed 27 May 2022).

Qi J., Yu Y., Wang L., Lui J. and Wang Y. (2017). An effective and efficient hierarchical K-means clustering algorithm [online]. Available at: https://journals.sagepub.com/doi/full/10.1177/1550147717728627 (accessed 30 March 2022).

Rajapaksha I. (2020). BERT Word Embeddings Deep Dive [online]. Available at: https://is-rajapaksha.medium.com/bert-word-embeddings-deep-dive-32f6214f02bf (accessed 27 January 2022).

Rubin V.L., Chen Y. and N.K. Conroy (2016). Deception Detection for News: Three Types of Fakes [online]. Available from: https://asistdl.onlinelibrary.wiley.com/doi/full/10.1002/pra2.2015.145052010083 (accessed 7 February 2022).

Sculley D. (2010). Web-scale k-means clustering [online]. Available from: https://dl.acm.org/doi/10.1145/1772690.1772862 (accessed 17 May 2022).

Sekaran U. and Bougie R. (2013). Research Methods for Business: A Skill Building Approach (6th ed.). West Sussex: John Wiley and Sons Ltd.

Snajder J. (2016). Social Media Argumentation Mining: The Quest for Deliberateness in Raucousness [online]. Available from: https://arxiv.org/abs/1701.00168 (accessed 07 January 2022).

Twitter (2022). Terms of Service [online]. Available at: https://twitter.com/en/tos (accessed 21 May 2022).

Tyrrell D.A. and Bynoe M.L. (1967). Cultivation of viruses from a high proportion of patients with colds [online]. Available from:

https://www.sciencedirect.com/science/article/abs/pii/S0140673666923646 (accessed 12 February 2022).
Umit R. (2022). Twitter Data in R Collection | Cleaning | Analysis [online]. Available from: https://resulumit.com/teaching/twtr_workshop.html#1 (accessed 7 April 2022).
UNDP.org (2020). UNDP: governments must lead fight against coronavirus misinformation and disinformation [online]. Available from: https://www.undp.org/content/undp/en/home/news-centre/news/2020/Governments_must_lead_against_coronavirus_misinformation_and_disinformation.html (accessed 21 June 2022).
Van der Maaten L. and Hinton G. (2008), Visualizing Data using t-SNE [online]. Available from: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (accessed 11 June 2022).
Velavan T.P. and Meyer C.G. (2020). The COVID-19 epidemic, [online]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7169770/#tmi13383-bib-0002 (accessed 7 March 2022).
Vlachos M., (2011), Dimensionality Reduction [online]. Available from: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_216 (accessed 9 April 2022).
Vialas V. (2018), A Natural Language Processing project for document clustering and topic extraction [online]. Available from: https://vitalv.github.io/projects/doc-clustering-topic-modeling.html (accessed 13 April 2022).
Wang W.Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection [online]. Available from: https://arxiv.org/pdf/1705.00648.pdf (accessed 17 February 2022).
World Health Organization (2022). WHO Coronavirus (COVID-19) Dashboard [online]. Available from: https://covid19.who.int/ (accessed 17 January 2022).

Yadav K. and Baria J. (2014). Mini-Batch K-Means Clustering Using Map-Reduce in Hadoop [online]. Available from: https://www.researchpublish.com/papers/mini-batch-k-means-clustering-using-map-reduce-in-hadoop (accessed 20 April 2022).

Yang S., Shu K., Wang S., Gu R.,Wu F. and Liu H. (2019). Unsupervised Fake News Detection on Social Media: A Generative Approach [online]. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/4508 (accessed 20 February 2022).

Yuthika A. (2018). Pros and Cons of K-means Clustering [online]. Available from: https://www.linkedin.com/pulse/pros-cons-k-means-clustering-aashima-yuthika/ (accessed 27 April 2022).

Zhang, C. et al., 2019. Detecting fake news for reducing misinformation risks using analytics approaches [online]. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0377221719304977 (accessed 07 November 2023).

Zhang X. and Ghorbani A.A. (2020), An overview of online fake news: Characterization, detection, and discussion [online]. Available from: https://www.sciencedirect.com/science/article/pii/S0306457318306794 (accessed 27 January 2022).