FAKE NEWS DETECTION SYSTEM BASED ON PROBABILISTIC SENTIMENT SCORE AND SENTENCE EMBEDDING

MAY ME ME HLAING

UNIVERSITY OF COMPUTER STUDIES, YANGON

JUNE, 2024

Fake News Detection System based on Probabilistic Sentiment Score and Sentence Embedding

May Me Me Hlaing

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

June, 2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..…… ……..……

Date May Me Me Hlaing

ACKNOWLEDGEMENTS

First and foremost, I would like to thank His Excellency, the Minister for the Ministry of Science and Technologies, for providing full facilities during the Ph.D. Course at the University of Computer Studies, Yangon.

Secondly, a very special gratitude goes to Dr. Mie Mie Khin, Rector of the University of Computer Studies, Yangon, for allowing me to develop this research and giving me general guidance during the period of my study.

I am also very greatful to Dr. Win Pa Pa, Professor and Course-coordinator of the Ph.D. 12th Batch, University of Computer Studies, Yangon, for her valuable advice, moral and emotional support in my research work.

I sincerely would like to express my greatest pleasure and the deepest appreciation to my supervisor, Dr. Win Lelt Lelt Phyu, Artificial Intelligence Lab, Professor, University of Computer Studies, Yangon. Without her excellent ideas, guidance, caring, and persistent help, this dissertation would not have been possible.

I deeply and specially thank the external examiner, Dr. Aung Nway Oo, Professor, University of Information Technology, Yangon, for his patience in critical reading, valuable suggestions and comments in the preparation of thesis.

I would like to express my respectful gratitude to Daw Aye Aye Khine, Professor and Head of English Department for her invaluable language assistance, which included pointing out proper usage not only in my Ph.D. course work but also in my dissertation.

My sincere thanks also go to all my respectful Professors for giving me valuable lectures and knowledge during the Ph.D. course work.

I also thank my friends from Ph.D. 12th Batch for providing support, care and true friendship along the way.

I am very much indebted to my family for always believing in me, for their endless love and support**.** They are always supporting and encouraging me during the years of my Ph.D. study. This accomplishment would not have been possible without them.

ABSTRACT

Analyzing of news truthfulness is a challenging problem in today's era because there is a massive information on the social networking sites (SNS) which turns out to be very difficult to manually analyze. Moreover, the impact of fake or negative news is tremendously huge to the internet users. In this complex field, scientists use sophisticated computer algorithms and neural network structures to examine and distinguish between the truthfulness of textual content that is distributed via various media channels. As a result, academic research related to filtering and banning fake news has been highly demanding since very recent years. Although there are some significant results and improvements made using different feature extraction methods and classification algorithms it still has some gaps to meet the important necessities to detect fake news because each method has some biases, variances and generalization errors. This research contributes to this area by using probabilistic sentiment score and sentence embedding, marks a significant advance forward in the accuracy of detecting fake news. It differs significantly from traditional approaches such as TF-IDF or bag-of-words (BOW) representation, which frequently ignore complex semantic and contextual nuances. The system first implements probabilistic sentiment model to get probabilistic sentiment score using TF-IDF, mutual information and logistic regression methods. Secondly, the system applies sentence embedding method to extract semantic and contextual feature vectors. The system finally uses Naïve Bayes and Support Vector Machine classifiers based on concatenated features (Probabilistic sentiment score and sentence embedding feature vector) for classification process. The system performs the experiments upon ISOT dataset and other fake news dataset from Kaggle. The effectiveness of the proposed method is remarkable 99% accuracy rate, which outperforms other models. Moreover, the results prove that the proposed concatenated feature is superior not only Naïve Bayes but also Support Vector Machine classifiers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

In today's digitalized world, the spread of information through social media[1] has developed extremely convenient. Anyone can create and disseminate information instantly with just a smartphone. The consumption of news through traditional media such as television and newspapers has significantly declined, giving way to the dominance of social media as the primary source of news for many people. However, this shift brings with it a major concern: the veracity of news on social media. Social media platforms are a combination of real and false information. While accurate news postures no problems, incorrect information can lead to social, economic, and political turmoil, particularly during time-sensitive events. The reliance on social media services has grown-up due to their plentiful advantages, such as facilitating social alertness, education, research, international connectivity, and real-time sharing of digital information. Consequently, the number of social media users has gradually increased over the years, playing a significant role in communication, establishing relationships, and expressing emotions. Significant issues include cyberbullying, hacking, and concerns concerning information privacy and security. However, one of the most pressing problems related to social media is the spread of fake news. Social media platforms have developed fast and available channels for spreading news internationally. Unfortunately, some individuals exploit this platform to disseminate false information for personal or social gain. Fake news can take the form of either misinformation (incorrect information) or disinformation (deliberately deceptive information)[2]. It typically purposes to deceive or mislead readers and shares similarities with spam messages, such limited set of words. These pieces of incorrect information often comprise emotionally charged content designed to influence the reader's opinion. Detecting false information effectively on social media postures a significant challenge that needs attention.

1.1 Fake News

Fake news is intentionally fabricated or misleading information presented as factual news[3], aiming to deceive readers, viewers, or listeners[4]. It can be completely fabricated or manipulate actual news to fit a specific narrative, employing false statements, doctored media[5], misleading headlines[6], or the omission of crucial information. The proliferation of fake news poses significant challenges by eroding trust in media, distorting public discourse, and undermining the reliable flow of information. Detecting fake news is a layered process that contains the analysis of the news contents to determine the truth of the news. The news could comprise information in various formats such as text, video, image, etc. Combinations of different types of data make the detection process difficult.

1.1.1 Impacts of Fake News

The common dissemination of fake news can have a significant impact on society. Since fake news is purposely fabricated, it can be applied for personal gain, financial or political purposes, and to tarnish the reputation of individuals or companies [7][8]. The magnitude of the impact caused by fake news is deeply influenced by aspects such as the timing and context in which the news is created, the social status of the person behind its creation, and the social media platform used for its dissemination. If measures to prevent the early spread of fake news are not executed, society may experience negative consequences.

Fake news is mainly intended to mislead readers, whereas a social media rumor refers to information that has not been verified for its accuracy at the time of its posting. Zubiaga et al. [9] defined a rumor as a circulating story that raises doubts about its veracity, appearing credible but hard to verify, often leading to scepticism and anxiety. Rumors can be true, incompletely true, or false, while fake news purposely presents false information in the guise of genuine news. Rumors[10] have the potential to spread misinformation or disinformation[11] . Detecting fake news can involve detecting similarities between fake news and rumors. Many methods have been proposed to detect rumors in social media. Typically, the problem of rumor detection is approached as a classification[12] problem such as a binary one.(rumor or non-rumor).

1.2 Problem Statements

Fake news influences people's perceptions. Spreading massive digital misinformation causes harmful effects from individuals to human society. Human readers difficult to accurately distinguish true from false information by just looking at these short pieces of information. Although there are some significant results and improvements made using different feature extraction methods and classification algorithms it still has some gaps to meet the important necessities to detect fake news because each method has some biases, variances and generalization errors.

Traditional text classification algorithms often rely on N-gram and Term Frequency Inverse Document Frequency (TF-IDF) as a technique to convert text into numerical vectors appropriate for machine learning classifiers[13][14][15]. Since text cannot be directly inputted into a classifier, TF-IDF is commonly used to represent the relative importance of words in a corpus based on their frequency of rate in documents. However, TF-IDF has limitations in capturing the position of words in the text, the context they belong to, semantic relationships, and occurrences across different documents, as it follows the bag-of-words (BoW) model[16]. In contrast, word embeddings offer a more nuanced approach by learning the positional representation of words in a vector space based on the surrounding words in the text. This contextual understanding becomes important for fake news detection, as the meaning of a word can vary depending on its context. Separating a word may yield a different meaning compared to considering it within a group of words in a paragraph.

While probabilistic sentiment scores generated by sentiment models are commonly used in text classification tasks like sentiment analysis, their application in fake news detection studies has never been used. Additionally, the current sentiment models often use the TF-IDF feature representation method, which can result in high-dimensional feature vectors. High dimensionality can cause computational inefficiencies, increased memory requirements, and the curse of dimensionality in machine learning techniques.

1.3 Motivation of the Research

Fake news detection is a serious challenge in today's information landscape, and developments in computational techniques can significantly contribute to improving the effectiveness of fake news detection systems. Firstly, incorporating probabilistic sentiment scores in fake news detection can offer valuable insights into the emotional tone and subjective opinions expressed within news articles. Sentiment analysis has been commonly used to analyze sentiments in texts, but its application in fake news detection remains relatively unexplored. By leveraging probabilistic sentiment scores, which go beyond simple positive/negative labels and capture the ambiguity related to sentiment predictions, a deeper understanding of the nuanced emotions can be gained present in news articles. This information can be leveraged to differentiate between reliable and possibly misleading news sources. Secondly, sentence embeddings offer a promising approach to capturing the contextual meaning and semantic relationships between words in a sentence. By representing sentences as continuous vector representations, sentence embeddings can capture the complex syntactic and semantic structures within news articles. Integrating sentence embeddings into fake news detection models can enhance their ability to classify subtle linguistic patterns and contextual cues that may specify the presence of misleading or fabricated information. Lastly, feature reduction techniques play a crucial role in mitigating the challenges posed by high-dimensional feature vectors. As fake news detection often includes processing large volumes of textual data, the dimensionality of feature representations can develop a computational bottleneck. Feature reduction techniques, such as dimensionality reduction or feature selection methods, can effectively reduce the dimensionality of the feature space while preserving the most informative features. This leads to improved computational efficiency, reduced memory requirements, and enhanced model performance. By combining probabilistic sentiment scores, sentence embeddings, and feature reduction techniques, this research purposes to develop a robust and efficient framework for fake news detection. The integration of these elements has the potential to uncover nuanced patterns, capture contextual information, and optimize the performance of fake news detection models. Ultimately, this research attempt seeks to contribute to the advancement of techniques that can effectively combat the proliferation of misinformation and promote the dissemination of reliable and trustworthy information.

1.4 The Objectives of the Research

The main objective of this research is to enhance the accuracy of fake news detection system using probabilistic sentiment scores and sentence embeddings. For increasing the accuracy of the fake news detection model, capture semantic understanding, provide robust feature representation, enable contextual analysis, enhance generalization, and fuse multiple features have been applied in the proposed model. The following are the other objectives:

Sentence embeddings offer a more compact and meaningful representation of sentences compared to traditional bag-of-words or TF-IDF approaches. They capture the semantic content of the sentence and provide a dense vector representation, which can be used as input for machine learning models. This enables more effective and robust feature representation for fake news detection.

Probabilistic sentiment scores and sentence embeddings can help improve the generalization capability of the fake news detection system. By incorporating probabilistic sentiment scores, the detection system aims to enhance the accuracy and granularity of fake news detection.

Feature reduction techniques aim to reduce the number of features in the model, resulting in faster training and prediction times. By eliminating or combining less informative features, the computational complexity of the sentiment model is reduced, making it more efficient.

1.5 Contributions of the Research

This research includes three main contributions which are described in the following paragraph:

The first contribution to this research employs the generation of probabilistic sentiment scores relies on the TF-IDF feature and a pre-existing sentiment model. However, when working with a big vocabulary, TF-IDF often produces a high-dimensional and sparse vector representation which can provide challenges in terms of memory usage and processing performance. The information gain approach for feature reduction is thus integrated into the sentiment model to speed up processing time.

Moreover, this research employs sentence embeddings as a text feature representation technique which offers a more compact and meaningful representation of sentences compared to traditional bag-of-words or TF-IDF approaches. Secondly, this research examines the impact of probabilistic sentiment scores on the system for identifying fake news. Sentence embeddings and probabilistic sentiment scores can greatly enhance the effectiveness of fake news identification. Probabilistic Sentiment scores give the likelihood that news is fake or real during the model training, whereas sentence embeddings capture the semantic content and contextual data.

The third contribution is that the system gets efficient accuracy for the detection of fake news using NB and SVM classifiers based on the proposed compound features (probabilistic sentiment score and sentence embedding features).

1.6 Scope and Limitations

The ISOT dataset, developed by the Information Security and Object Technology (ISOT)[17] Research Lab, was used in this study. The dataset contains both real news articles and intentionally fabricated or misleading news articles, commonly referred to as fake news. The articles cover a range of topics and domains, reflecting the diversity of news sources. Each news article in the dataset is associated with a binary label indicating whether it is real or fake. Including datasets from more recent periods, as well as news articles from previous periods, would improve the model's ability to generalize and handle recently gathered data points. While sentence embeddings are typically employed for training deep learning algorithms[18], this study applied these approaches to a machine learning algorithm[19]. The hyperparameters employed in the LSVM model[20] were set to default features since earlier research [21] did not specify the parameters used to reproduce their results, despite achieving the same accuracy using the 1-gram TF-IDF text representation technique. Any attempts to correct the hyperparameters only resulted in reduced accuracy across all models tested, thus they were left unchanged.

1.7 Organization of the Research

The arrangement of this research is as follows: in Chapter One, the introduction of the definition of fake news and their impacts on society, problem statements, motivation,

objectives, contribution, and scope and limitation of the thesis are described. In Chapter Two, the literature reviews and associated work, as well as some existing approaches that were also evaluated in previous research dealing with the subject are presented. In Chapter Three, background theory, examinations of all prior research, and implementations in the fields of fake news detection, word embeddings, sentence embeddings, text classification, and TF-IDF are explained in detail in which there are research gaps discovered. The review process identifies weaknesses, which then serve as the basis for framing a research topic. This chapter also covers data preprocessing, logistic regression, feature reduction, probabilistic sentiment scores, and advanced methods of detecting fake news. In Chapter Four, the comprehensive system architecture proposed for the fake news detection system is described. The chapter initiates by providing a detailed description of the datasets used, emphasizing the importance of data understanding as a main objective in this section. Understanding data helps to comprehend why these specific data preparation and cleaning procedures were chosen for usage for specific datasets. This section also includes discussions of feature extraction such as sentence embedding with InferSent, probabilistic sentiment models, and combined features. Finally, Fake news detection using SVM and Naive classification with our proposed features is extensively displayed. In Chapter Five, the evaluation of results from experiments using SVM and NB are described. This section also discusses the effect of PSS on those classifiers for detecting fake news. Finally, a summary of the research and future extension of research are described in Chapter Six.

CHAPTER 2
LITERATURE REVIEWS

In this section, the focus is on documenting the previous research conducted in the field of Fake news detection, mainly the studies that utilized the Linear Support Vector Machine and Naïve Bayes model. TF-IDF, word embedders, and sentence embedders were also discussed for text classification purposes. This is followed by specifying the state of the art and the research gaps that were identified.

2.1 Fake News Detection

The effort of identifying and classifying news items or textual information as either real or fake is referred to as fake news detection." This problem falls within the realm of natural language processing. In order to reduce the spread of fake information and make it easier for people to get accurate information, the purpose of this effort is to develop algorithms that are capable of identifying and flagging items of fake news on their own. The dissemination of fake news has garnered increasing attention, particularly due to its use in spreading political propaganda, influencing elections, and harming individuals or groups. Highly sophisticated applications, or bots, are organized in networks to massively propagate fake news across social media platforms, utilizing various formats such as text, images, audio, or video files. Detecting fake news poses significant challenges, as it involves navigating the delicate balance between freedom of speech and the need to combat misinformation. Manual fact-checking is limited by its inability to keep pace with the vast volume of fake news spread on social media. Automation has emerged as a solution, employing techniques such as artificial intelligence, natural language processing, and blockchain to verify news content and detect fake accounts or campaigns [22][23]. However, the effectiveness of these algorithms remains a concern, particularly regarding accuracy and potential biases.

Research indicates that human behavior contributes more to the spread of fake news than automated bots, emphasizing the importance of increasing societal resilience and media literacy. Initiatives to raise awareness and enhance media literacy can positively

impact data protection by empowering consumers to critically evaluate media messages and safeguard their personal data.

However, there are negative foreseen impacts on data protection as well. Lack of transparency and legal basis in fake news detection algorithms raises concerns about individuals' rights to access, correct, and delete their personal data. Moreover, algorithmic inaccuracies and the increase in automated decision-making[24] without sufficient human oversight pose risks of biased results and limited accountability.

Overall, addressing fake news requires a multi-faceted approach that combines technological solutions with efforts to empower individuals and enhance media literacy, while also ensuring transparency, accountability, and protection of personal data in the process.

2.2 Natural Language Processing

Natural Language Processing (NLP)is a subfield of artificial intelligence (AI) and computational linguistics that focuses on the interaction between computers and human language[25][26][27]. It involves developing algorithms and techniques to enable computers to understand, interpret, and generate human language in a meaningful way. NLP encompasses a wide range of tasks and applications related to natural language understanding and generation. The techniques employed in NLP often involve various components, including tokenization[28] (breaking text into individual words or tokens), part-of-speech tagging[29] (assigning grammatical labels to words), syntactic parsing[30] (analyzing sentence structure), semantic analysis[31] (deriving meaning), and statistical modelling[32] (using machine learning algorithms to make predictions or extract patterns from data). NLP has several applications in areas such as information retrieval[33], chatbots[34], virtual assistants[35], language translation[36], sentiment analysis, social media analysis, healthcare[37], customer support[38], fake news detection, and many more to enable computers to comprehend and communicate with humans in natural language.

2.3 Machine Learning and Natural Language Processing

Machine Learning and Natural Language Processing (NLP) are crucial aspects of Artificial Intelligence (AI), gaining significant attention in recent years. NLP enhances AI systems, enabling them to understand and interact with users in a more intuitive manner. Machine learning empowers AI systems to learn from data and make predictions, significantly improving their performance compared to traditional hardcoded algorithms.

Deep Learning, a subset of machine learning, centered around Artificial Neural Networks (ANN)[39], has shown remarkable success, particularly in NLP tasks. The flexibility of deep learning architectures has contributed to their widespread adoption and success in various applications. NLP enables computers to understand and process human languages, enhancing user-friendliness across numerous applications, from controlling electronic devices to interacting with complex systems. While NLP simplifies user interactions, it involves complex processing facilitated by machine learning algorithms working behind the scenes.

2.4 Role of NLP in Machine learning

The utilization of Natural Language Processing (NLP) in machine learning enhances machines' comprehension of human language. This entails several steps such as deciphering word structure, sentence structure, and meaning. Machine Learning serves as a valuable tool in each of these steps, facilitating a smoother language understanding process. Essentially, it involves teaching machines to improve their comprehension and response to natural speech and writing patterns. Below are some ways in which NLP and machine learning collaborate to enhance language understanding.

2.4.1 Morphological Analysis

At this stage, the computing system receives data in the form of binary code, which is then converted into characters using ASCII code. Tokenization, the process of identifying words and sentences, is the primary task in morphological analysis. Various machine learning and deep learning algorithms, such as Support Vector Machine and Recurrent Neural Network[40], are employed for tokenization. Following tokenization, affixes in sentences complicate matters for machines, necessitating their removal through techniques

like stemming or lemmatization. Algorithms like random forest[41] and decision tree[42] are effective in performing stemming tasks.

2.4.2 Syntactic Analysis

This phase involves checking whether a given sentence adheres to the grammar rules of a language. Part-of-speech tagging[43] precedes the syntactic parsing process. Machine learning and deep learning algorithms like random forest and recurrent neural network are commonly used for this purpose, with algorithms like K-nearest neighbor[44] also being employed for syntactic parsing.

2.4.3 Semantic Analysis

Semantic analysis involves identifying word meanings using dictionaries. However, ambiguity arises when the same word has multiple meanings based on the sentence context. Resolving this ambiguity, known as Word Sense Disambiguation, is a critical task. Classical classification problems like word sense disambiguation have been addressed using machine learning algorithms such as random forest, gradient boosting, and decision trees. Deep learning algorithms like recurrent neural network and convolution neural network have shown promising results in recent times.

2.4.4 Discourse Analysis

Instances where pronouns or subjects/objects are referred to outside the current context pose challenges for semantic analysis. Reference resolution tackles this problem, employing both machine learning and deep learning algorithms.

2.4.5 Pragmatic Analysis:

Sentences often convey implied meanings that go beyond literal interpretation. Detecting such deeper meanings, such as sarcasm, presents a significant challenge. Various machine learning and deep learning algorithms have been explored for sarcasm detection and pragmatic analysis in general, with varying degrees of success.

2.5 The applications of Natural Language processing

In the realm of natural language processing (NLP) within machine learning, deep learning algorithms have emerged as pivotal components across various applications. Recent research has seen a resurgence of interest in these fields due to the ease of implementing machine learning and deep learning algorithms, particularly deep learning techniques.Consequently, a wide array of deep learning methods, including Deep Neural Networks, Autoencoders[45], Restricted Boltzmann Machines[46], Recurrent Neural Networks, and Convolutional Neural Networks, have been extensively explored to achieve high accuracy in diverse NLP applications.Among these, Recurrent Neural Networks, along with their variants such as Long Short Term Memory and Gated Recurrent Unit, as well as Convolutional Neural Networks and their derivatives like Recurrent Convolutional Neural Networks and Regional Convolutional Neural Networks, have been subjected to thorough research to yield positive outcomes in various NLP applications.

2.5.1 Sentiment Analysis

Sentiment analysis is a crucial aspect of analyzing user feedback and understanding their sentiments or opinions towards a particular product, service, or topic. This analysis plays a pivotal role in Customer Relationship Management (CRM)[47], as even a single negative review or comment can have a substantial impact on how a product or service is perceived by potential customers and the general public.

In recent years, there has been a significant increase in the adoption of deep learning techniques for sentiment analysis tasks. Deep learning, a subset of machine learning, involves the use of neural networks with multiple layers to automatically learn and extract complex patterns and features from data. These deep learning methodologies have revolutionized sentiment analysis by providing more accurate and nuanced insights into user sentiments.

What particularly noteworthy is the development of new deep learning models and algorithms that are specifically tailored for sentiment analysis tasks. Researchers and practitioners in the field have dedicated considerable effort to designing deep learning architectures that can effectively capture the subtle nuances and nuances of human language, emotions, and expressions found in textual data.

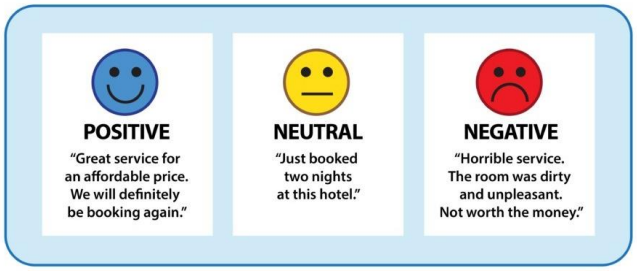


Figure 2.1 Example of Sentiment Analysis

These specialized deep learning techniques for sentiment analysis delve into various aspects, such as sentiment polarity (positive, negative, neutral), emotion detection[48] (e.g., happiness, sadness, anger), and opinion mining (identifying subjective opinions and viewpoints). By leveraging the power of deep learning, sentiment analysis models can analyze large volumes of text data, including social media posts, customer reviews, and feedback surveys, to extract valuable insights and sentiments.Overall, the increasing adoption of deep learning methodologies for sentiment analysis reflects the ongoing advancements and dedication within the field to develop more sophisticated and accurate tools for understanding and interpreting human sentiments and opinions.

2.5.2 Chatbot

Chatbot systems, also known as conversational agents, are interactive interfaces designed to engage users in conversations, typically through text or voice interactions. The widespread adoption of personal assistants like Amazon's Alexa and Google Assistant has brought chatbot systems into the mainstream, showcasing how users can effortlessly interact with them to perform various tasks and gather information.However, despite the apparent simplicity of interaction, creating a fully functional chatbot system that can effectively replace a human agent is an incredibly complex and challenging endeavor. It requires specialized expertise in areas such as Natural Language Understanding[49] (NLU) and Natural Language Generation[50] (NLG) to ensure that the chatbot can comprehend user inputs accurately and generate meaningful responses.

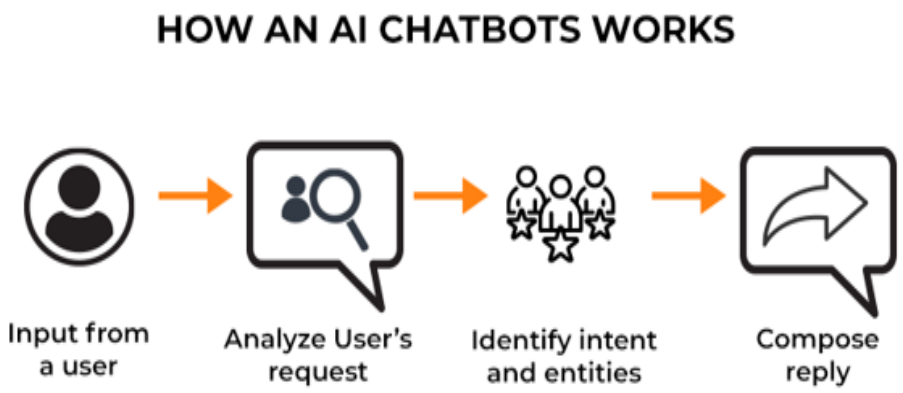


Figure 2.2 Example of Chatbot Application

Modern frameworks and platforms have emerged to streamline the development process of chatbot systems. Examples of these frameworks include Google's DialogFlow, IBM's Watson AI, and Amazon's Alexa AI. These platforms offer developers a range of tools, APIs, and pre-built components that simplify the implementation of advanced NLU and NLG capabilities within chatbots.One of the key advantages of using these modern frameworks is their integration with sophisticated deep learning architectures. Deep learning techniques, such as neural networks, enable chatbots to learn from large volumes of data and improve their language understanding and generation capabilities over time. These frameworks also often incorporate proprietary algorithms and models that enhance the chatbot's ability to handle complex dialogues and provide more contextually relevant responses.

In contrast, while chatbot systems have become increasingly popular and accessible thanks to advancements in technology and the availability of modern development frameworks, building a truly effective and intelligent chatbot still requires a deep understanding of NLU, NLG, and the utilization of sophisticated deep learning techniques offered by platforms like DialogFlow, Watson AI, and Alexa AI.

2.5.3 Question Answering System

In recent times, the lines between dialogue systems and question answering systems[51] have become increasingly blurred, thanks to the evolution of question

answering systems that now incorporate conversational elements. This integration has led to overlapping functionalities, where chatbot systems often perform question answering tasks, and vice versa. As a result, contemporary research efforts in developing chatbot systems are more inclined towards incorporating robust question answering capabilities.

A typical question answering system comprises three fundamental components: Question Processing, Information Retrieval, and Answer Processing. Each of these components plays a crucial role in ensuring the system can effectively respond to user inquiries with accurate and relevant information. Machine Learning (ML) and Deep Learning (DL) techniques have been instrumental in enhancing the capabilities of these components. Question Processing is a pivotal stage in a question answering system, where the system must understand the user's query accurately to retrieve the appropriate answer. This aspect has garnered significant research attention, with a focus on improving question comprehension to facilitate better answer retrieval. Question processing is often approached as a classification problem, where the system categorizes questions into different types or categories based on their structure and semantics. Deep learning techniques, such as neural networks, have been extensively explored in research endeavors to enhance question classification accuracy.

Information Retrieval is another critical component, where the system retrieves relevant information or knowledge from a structured or unstructured data[52] source to formulate an answer. This process involves techniques such as keyword extraction[53], semantic analysis, and document retrieval[54], all of which can benefit from machine learning algorithms to improve precision and recall in retrieving information.Finally, Answer Processing involves generating a coherent and relevant response based on the retrieved information. This phase may include natural language generation techniques, where ML and DL models are trained to generate human-like responses that are grammatically correct and contextually appropriate.

Overall, the integration of machine learning and deep learning techniques has revolutionized question answering systems, enabling them to handle complex queries, understand nuances in language, and provide accurate and informative responses, thereby advancing the capabilities of modern chatbot systems.

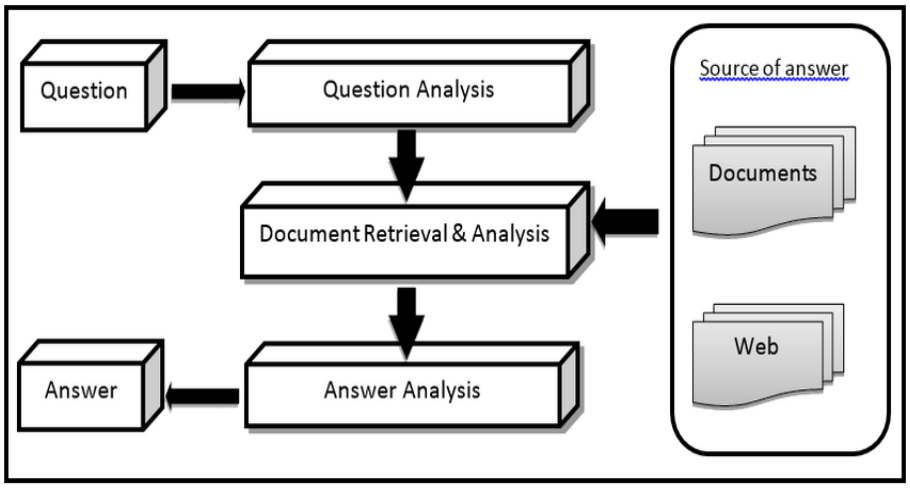


Figure 2.3 Example of Question-Answering Application

2.5.4 Information Retrieval

Information Retrieval Systems (IRS) are pivotal applications within the realm of Natural Language Processing (NLP), serving the critical function of fetching pertinent information from vast data repositories. These systems play a foundational role in various technological applications, such as chatbots and question answering systems, by enabling them to provide accurate and relevant responses to user queries.

While the frequency method stands as a fundamental approach to information retrieval, relying on keyword frequency to determine data relevance, modern IRS have evolved to incorporate more sophisticated techniques. These advanced systems not only analyze the query input but also delve into extensive data sets to retrieve only the most pertinent information. This intricate process is facilitated by leveraging deep learning techniques, which enable IRS to understand context, semantics, and nuances within the data, leading to more accurate and context-aware retrieval results.

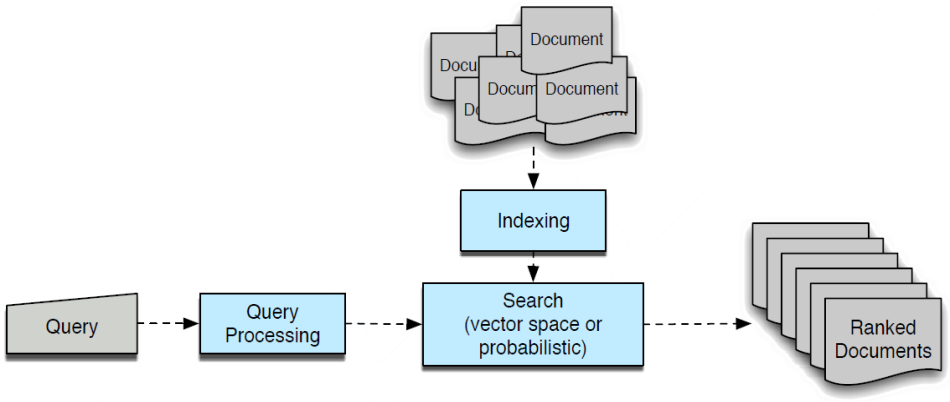


Figure 2.4 Workflow of Information Retrieval System

By integrating deep learning methodologies into Information Retrieval Systems, organizations and researchers can achieve enhanced performance, improved accuracy, and a deeper understanding of user intent, ultimately enhancing the overall user experience and enabling more efficient knowledge discovery.

2.5.5 Machine Translation

A machine translation system is designed to accomplish the intricate task of translating text from one language to another, aiming for seamless communication across linguistic barriers with minimal or no human intervention. Prominent platforms like Google Translate serve as prime examples of such machine translation systems, showcasing the advancements made in automating the translation process.

The complexity of machine translation arises from the fact that mere word-for-word translation is often inadequate. This is because different languages may have varying sentence structures; for instance, English typically follows the Subject-Verb-Object format, whereas languages like Hindi may utilize the Subject-Object-Verb structure. Moreover, there are numerous linguistic rules, nuances, and cultural contexts that must be considered, further complicating the translation task.

Within the domain of machine translation, particularly in the intersection of natural language processing (NLP) and machine learning, researchers have extensively explored various techniques to enhance translation quality. One such technique is the Recurrent Neural Network (RNN), along with its derivatives like Long Short Term Memory (LSTM)

and Gated Recurrent Unit (GRU), including their bidirectional forms. These neural network architectures have been subject to extensive experimentation due to their ability to retain contextual information, which is crucial for producing accurate and meaningful translations.

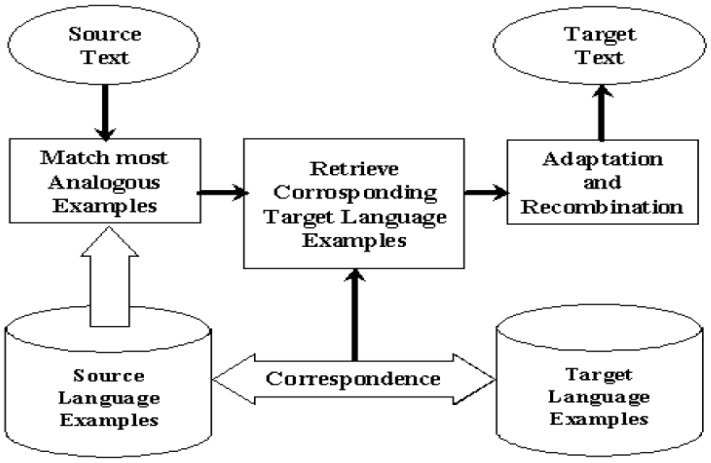


Figure 2.5 Machine Translaiton Architecutre

Additionally, Convolutional Neural Networks (CNNs) have also been explored in the context of machine translation, albeit with varying degrees of success. CNNs are known for their effectiveness in capturing spatial patterns in data, and their application in translation tasks has shown promise, particularly in handling specific types of linguistic structures and syntactic features.

Overall, the continuous advancements in neural network architectures and machine learning algorithms have significantly contributed to the improvement of machine translation systems, enabling them to achieve higher levels of accuracy, fluency, and semantic understanding, thus bridging language barriers and facilitating global communication.

2.6 Text Classification Methods

The process of text classification can be approached manually or automatically. Manual classification involves human annotators who categorize text based on its content, which, while accurate, is time-consuming and costly. Automatic text classification, on the

other hand, utilizes machine learning, natural language processing (NLP), and other AI techniques to categorize text more efficiently and accurately.

Automatic text classification can be achieved through rule-based systems, machine learning-based systems, or hybrid systems. Rule-based systems classify text using handcrafted linguistic rules, which can be effective but require deep domain knowledge and are challenging to maintain. Machine learning-based systems, on the other hand, learn classification patterns from labeled training data, making predictions based on observed associations between text features and categories. This approach is more scalable and accurate, especially for complex tasks, although it requires substantial training data and computational resources.

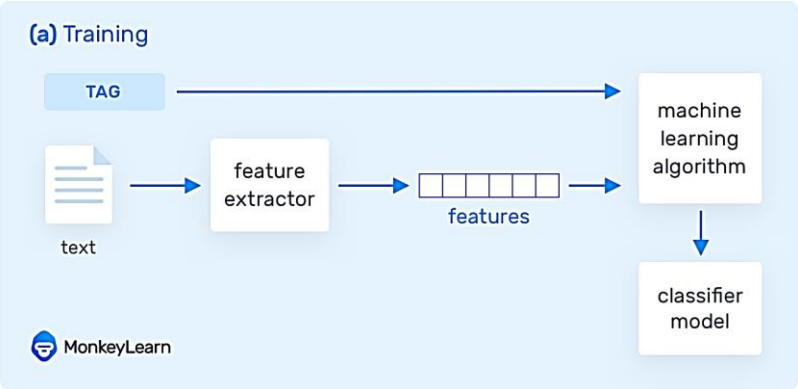


Figure 2.6 Training Pipeline for Text Classification Model

Some popular machine learning algorithms for text classification include the Naive Bayes family of algorithms, support vector machines (SVM), and deep learning models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Naive Bayes algorithms are known for their simplicity and efficiency, while SVMs are powerful for multi-dimensional classification tasks. Deep learning models offer high accuracy and performance but require large amounts of training data.

Hybrid systems[55] combine machine learning classifiers with rule-based systems to improve classification results further. These systems leverage the strengths of both approaches and can be fine-tuned with specific rules to address any shortcomings in the base classifier's performance. Overall, automatic text classification offers a faster, more

cost-effective, and scalable solution compared to manual methods, making it ideal for various applications in natural language processing and information retrieval.

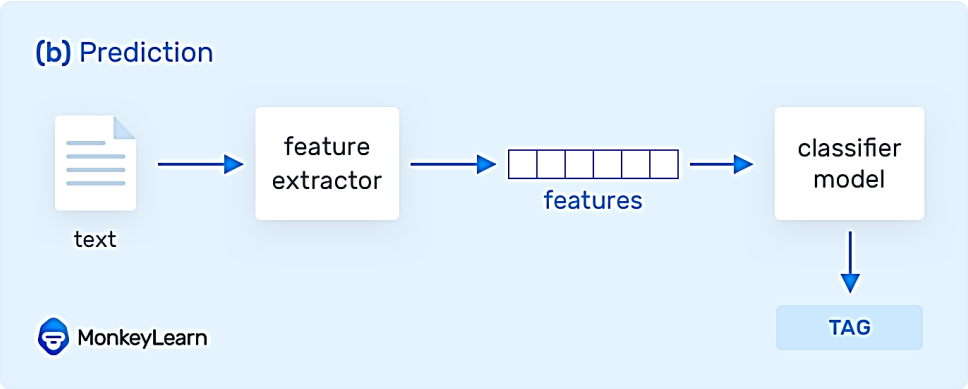


Figure 2.7 Testing Pipeline for Text Classification Model

Presented below are some of the most widely used approaches in this text classification field:

Naive Bayes Classifier:Naive Bayes, a probabilistic algorithm founded on Bayes' theorem, works under the assumption of conditional independence among features (words) given the class label. Despite its "naive" assumption, Naive Bayes is simple, efficient, and works well for many text classification tasks, especially when there is limited training data.

Support Vector Machines (SVM):SVM[56] is a popular machine learning algorithm used for both binary and multiclass classification tasks. In the context of text classification, SVM is used to find the optimal hyperplane that effectively separates the data points representing different classes within a high-dimensional space.

Logistic Regression: Logistic regression is another widely used algorithm for binary text classification tasks. It constructs a model to estimate the probability of a text belonging to a specific class using a logistic function. Moreover, it can be extended to handle multiclass classification with techniques such as One-vs-Rest or One-vs-One.

Random Forest[57] is a powerful ensemble learning technique widely employed in machine learning for making predictions across various domains. This method is particularly effective for tasks such as classification. At its core, Random Forest utilizes the concept of decision trees, where multiple trees are created and then combined to arrive at a final prediction.

In the context of text classification, Random Forest can be harnessed to construct a "forest" of decision trees based on different features, typically words or tokens extracted from the text data. Each decision tree in the forest is trained on a random subset of features and data samples from the training set. This randomness in feature selection and sample selection helps in reducing overfitting and improves the generalization ability of the model.

Once the Random Forest is trained, it aggregates the predictions from each decision tree to arrive at a final prediction for a given input. This aggregation process can be done through various methods such as voting (for classification tasks) or averaging (for regression tasks).

The strength of Random Forest lies in its ability to handle high-dimensional data, handle missing values, and provide feature importance scores, making it a versatile and reliable choice for text classification tasks. Its ensemble approach helps mitigate the biases and errors associated with individual decision trees, leading to more robust and accurate predictions.

XGBoost:XGBoost[58], short for eXtreme Gradient Boosting, is a powerful supervised machine learning algorithm renowned for its effectiveness in handling classification and regression tasks. This algorithm operates by constructing a series of decision tree models, referred to as base learners, in a sequential manner. Each base learner contributes essential estimations that collectively enhance the algorithm's predictive capabilities. By amalgamating these estimates from the base learners, XGBoost models are adept at making accurate and reliable decisions.

Consider a scenario involving a dataset comprising rows of speeches categorized as 0 for hate speech and 1 for neutral speech. In this case, the XGBoost classification model is trained using this dataset, with the option to specify the desired number of estimators, which corresponds to the number of base learners or decision trees. Once the text dataset is trained using XGBoost, a new test dataset containing different inputs can be fed into the model for making predictions.

KNN: KNN[59], which stands for K Nearest Neighbour, is a supervised machine learning algorithm widely utilized for classification tasks. This algorithm operates by identifying the K nearest data points (neighbors) in the training dataset that are most similar to the new data point being classified. By leveraging the similarities and characteristics

shared among neighboring data points, KNN makes predictions based on the majority class or group within its nearest neighbors.

The KNN algorithm determines the nearest neighbors by assessing the closeness and proximity among the features of the training data. This proximity is often measured using metrics such as Euclidean distance or Manhattan distance. The model is trained on this proximity information so that when new data is introduced to the model, it can efficiently determine its class or group based on the majority class among its K nearest neighbors.

In the image provided, you can observe the process of classifying new data using the KNN model. The new data point, represented as a circle, is assigned to category 1 after passing through the KNN algorithm, as it is closer and more similar to the data points belonging to category 1 in the training dataset. This exemplifies how the KNN algorithm leverages the concept of proximity and similarity to make accurate predictions for new data points.

2.7 Feature Engineering in NLP

In natural language processing (NLP), a feature refers to a measurable characteristic or property extracted from raw text data to represent it in a structured and numerical format. Features play a fundamental role in NLP tasks as they enable machine learning algorithms to process and comprehend language efficiently. Feature Engineering[60] is the core of any Machine Learning model. The performance of the model and predictive accuracy deeply rely on the application of various feature engineering techniques. It can be considered the 'art' of devising valuable features from existing data, taking into account the target to be learned and the machine learning model used. The process involves transforming data into formats that create a more significant connection to the underlying target for learning When executed right, feature engineering can enhance the value of existing data and improve the overall performance of machine learning models. On the other hand, using bad features may require building more complex models to achieve similar levels of performance. Feature engineering mainly serves two key objectives:

- Feature engineering includes preparing the input dataset in a format suitable for a specific model or machine learning algorithm.

- Feature engineering helps in improving the performance of machine learning models magically.

Most classic machine learning algorithms are unable to process raw text directly. So, it becomes essential to conduct feature extraction from the raw text, converting it into numerical features that can be utilized by machine learning algorithms.

2.7.1 Bag of Words (BoW)

This is one of the simplest vector-space models used for unstructured text. The vector space model is a mathematical approach for representing unstructured text (or any other data) as numeric vectors. Each dimension of the vector corresponds to a distinct feature characteristic In this model, text documents are represented as numeric vectors using the bag of words model[61], where each dimension represents a specific word from the corpus, and the value can signify its frequency in the document, occurrence (denoted by 1 or 0), or even weighted values. The term "bag of words" derives from the concept that each document is represented just as a 'bag' containing its words, without considering word order, sequences, or syntax.

Drawbacks of using a BOW model: when new sentences contain previously unseen words, it leads to an increase in the vocabulary size, beginning the length of the vectors to expand correspondingly. Consequently, the vectors would contain many 0s, resulting in a sparse matrix (which is generally undesirable). Furthermore, in this approach, any information about the sentence structure or word order within the text is not retained.

2.7.2 Term Frequency and Inverse Document Frequency (TF-IDF)

The TF-IDF model attempts to address this issue by integrating a scaling or normalizing factor into its computation. TF-IDF stands for Term Frequency-Inverse Document Frequency which is calculated using a combination of two metrics: term frequency (tf) and inverse document frequency (idf).

$$TF_{(W,D)} = \frac{n_{W,D}}{Number\ of\ Terms\ in\ the\ Document} \qquad \text{Equation (2.1)}$$

$$IDF_{(W,D)} = \frac{number\ of\ documents}{Number\ of\ documents\ with\ term\ 'w'} \qquad \text{Equation (2.2)}$$

Mathematically, TF-IDF can be defined as tfidf = tf x idf, where the term tfidf (w, D) represents the TF-IDF score for word w in document D. The term tf (w, D) reflects the term frequency of word w in document D, as generated using the Bag of Words model.

The term idf (w, D) denotes the inverse document frequency for the term w, which may be calculated as the log transform of the total number of documents in the corpus C divided by the document frequency of the word w. The document frequency of word w corresponds to the frequency of documents in the corpus where the word w occurs.

When compared to the raw Bag of Words model values, the TF-IDF-based feature vectors for each of our text documents provide scaled and normalized values. Bag of Words just generates a set of vectors providing the count of word occurrences in the text (reviews), but the TF-IDF model includes information on both the most important and less significant words. Bag of Words vectors are simple to comprehend. Nevertheless, in machine learning models, TF-IDF typically outperforms.

2.8 Document and Sentence Embedding

In Natural Language Processing (NLP), sentence embedding involves converting a sentence into a numerical representation as a vector of real numbers, capturing its semantic meaning. This representation enables comparisons of sentence similarity by measuring the distance or similarity between these vectors. Techniques such as the Universal Sentence Encoder (USE) employ deep learning models trained on extensive text corpora to generate these embeddings, which are useful in various tasks like text classification, clustering, and similarity matching [62].

2.8.1 Embedding

Embeddings provide a solution to the challenges posed by traditional techniques like TF-IDF in representing textual data for machine learning models. While TF-IDF can result in high-dimensional representations that increase the model's error as the number of features grows, embeddings offer low-dimensional, distributed representations. These representations map words or sentences to vectors of real numbers, ensuring that words or sentences with similar meanings have similar embeddings. Thus, embeddings enable the

numerical representation of textual data in a way that captures semantic similarities effectively, facilitating improved model performance in tasks such as identifying fake news. Embeddings not only transform words or text into numerical representations, but also capture and integrate its semantic and syntactic content.

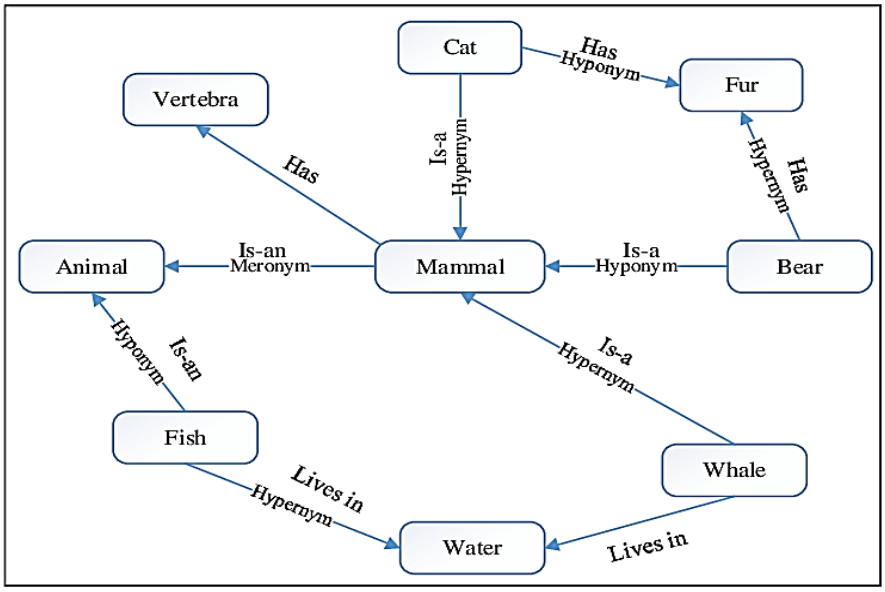


Figure 2.8 Semantic Relations Like the Relation between Words

Embeddings are representations of words and text that are spread out in a continuous vector space. They may help with tasks such as finding related meanings, grouping similar items, making suggestions, analysing emotions, answering questions, or removing duplicates. Embeddings are representations of words and text that are spread out in a continuous vector space. They may help with tasks such as finding related meanings, grouping similar items, making suggestions, analysing emotions, answering questions, or removing duplicates [63].

2.8.2 Word Embedding

Word embeddings are vectorized representations of words and are considered a crucial advancement in the field of Natural Language Processing (NLP). Let's examine some of the primary algorithms that are now being utilized.

2.9 Word2vec Method

Word2vec[64] [65]is a computational model used to represent words as numerical vectors in order to capture their semantic meaning. Since its establishment in 2013, Word2vec has gained significant popularity and is now extensively used in both academic research and commercial applications. The concept is founded on the notion that it is feasible to anticipate a word by considering its context, namely the words around it. This assumption is grounded on the belief that the meaning of a word can be deduced from the other words it is often associated with. Word2vec employs two architectures, namely continuous bag-of-words (CBOW) and Skip-gram, to generate a distributed representation of words. In CBOW, the current word is predicted based on a window of surrounding context words, whereas in Skip-gram, the main word is used to predict the context words.

This is where Word Embedding techniques such as Word2Vec, Continuous Bag of Words (CBOW), Skipgram[66], and others come into play. These techniques purpose to capture the semantic relationships between words and represent them in a dense vector space, offering a more nuanced and context-aware representation of words in text data. The primary goal of word embeddings is to capture semantic relationships between among words, ensuring that similar words are located closer to each other in the vector space. This representation allows machine learning models to better understand the meaning and context of words, which can lead to improved performance on various NLP tasks. The most common approach for learning word embeddings is the Word2Vec algorithm, which introduced the concept of distributed word representations. Word2Vec uses a shallow neural network with either the Continuous Bag of Words (CBOW) or Skip-gram architecture to learn word embeddings from large text corpora.

2.9.1 Continuous Bag of Words (CBOW) Model

The CBOW model[67] architecture attempt to predict the current target word (the center word) based on the surrounding source context words. For instance, in a simple sentence like "the quick brown fox jumps over the lazy dog," this process involves forming pairs of (context_window, target_word). Considering a simple sentence, “the quick brown fox jumps over the lazy dog”, this can be pairs of (context_window, target_word) where if we consider a context window of size 2, we have examples like ([quick, fox], brown), ([the,

brown], quick), ([the, dog], lazy) and so on. As a result, the model attempt to predict the target_word' based on the words within the 'context_window'.

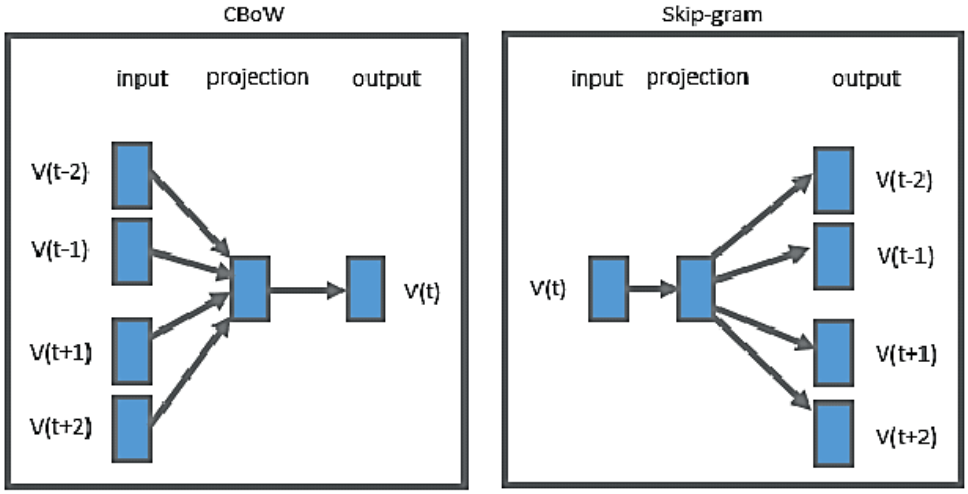


Figure 2.9 CBow and Skip-gram Architecture

2.9.2 The Skip-gram Model

The Skip-gram model design typically attempts to achieve the opposite of the CBOW model does. Given a target word (the center word), it attempts to predict the source context words (surrounding words). Considering our previous example sentence: "the quick brown fox jumps over the lazy dog." We get pairs of (context_window, target_word) if we use the CBOW model, and for a context window of size 2, we get instances like ([quick, fox], brown), ([the, brown], quick), ([the, dog], lazy), and so on. Given that the purpose of the skip-gram model is to predict the context from the target word, the model typically reverses the contexts and targets and attempts to predict each context word from its target word. As a result, the objective becomes predicting the context [quick, fox] given the target word'brown' or [the, brown] given the target word'quick,' and so on. As a result, the model attempts to forecast the words within the context window based on the given target word.

2.9.3 Global Vector (GloVe)

GloVe, short for Global Vectors for Word Representation[68], is a word embedding approach that was created by Stanford University. One benefit of this method over Word2Vec is that it takes into account not only the local statistics (contextual information of the words), but also the global statistics (word co-occurrence) from the whole text

corpus. GloVe uses global text-level co-occurrence information to generate vector representations of words. This element is significant since word-word co-occurrences may contain rich semantic information. For instance, in a vast collection of texts, the term "solid" is more prone to appearing along with "ice" rather than "steam". Conversely, the word "gas" is likely to co-occur more often with "steam" than with "ice".

2.9.4 FastText

Facebook created FastText[69] with the aim of using the basic structure of words to improve vector representations, giving it an important edge over competing models.Word2Vec and GloVe produce word embeddings that are limited to processing just the words they have been trained on, finding them ineffective in handling words that were not included in their training data. On the other hand, FastText has the capability to produce word vectors for unknown words, creating embeddings for words that have not been seen before.

Figure 2.10 Character n-grams for the word “eating”

2.9.5 Sentence Embedding

Instead of dealing with individual words, sentence embedding allows us to work directly with entire sentences, especially beneficial when dealing with large texts where word-level analysis might be insufficient. For instance, understanding the connection between concepts like "crowded places" and "busy cities" requires capturing the context and semantic nuances of entire sentences, which traditional word embedding methods may struggle with.

Sentence embedding models aim to encapsulate the semantic essence of a sentence in a fixed-length vector, going beyond the limitations of traditional Bag-of-Words (BoW) representations. Methods for generating sentence embeddings include averaging word

embeddings within a sentence, utilizing pre-trained models like BERT for context-aware embeddings, and neural network-based approaches such as Skip-Thought vectors and InferSent, which are trained to predict surrounding sentences.

Several libraries and tools, such as Doc2Vec, SentenceBERT, InferSent, and the Universal Sentence Encoder, facilitate the generation and use of sentence embeddings, providing valuable resources for enhancing the understanding of language by machines.

Doc2Vec, an extension of Word2Vec, extends word embedding techniques to generate embeddings for entire documents or paragraphs. This allows encoding semantic meaning and context into fixed-length vectors for efficient Natural Language Processing (NLP) tasks.

There are two main architectures in Doc2Vec:

Distributed Memory (DM): Takes context words and a document ID to predict the target word, incorporating the document vector and word vectors.

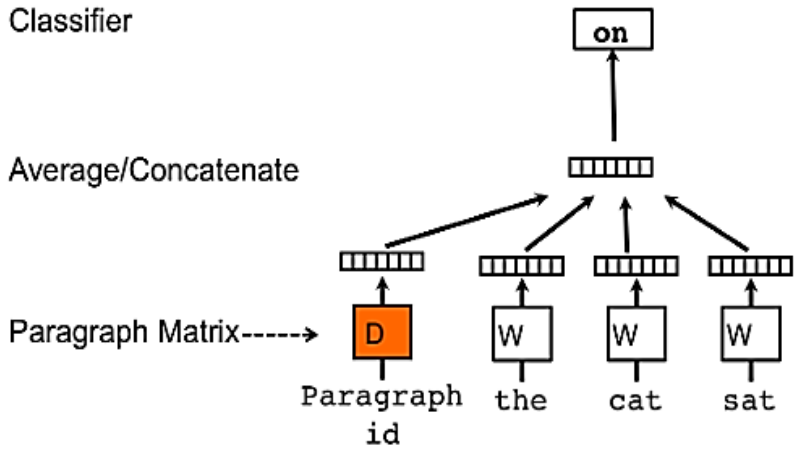


Figure 2.11 Distributed Memory (DM) Doc2Vec

Distributed Bag of Words (DBOW): Ignores word order, using only the document ID to predict the target word, inferring the document vector without considering context.

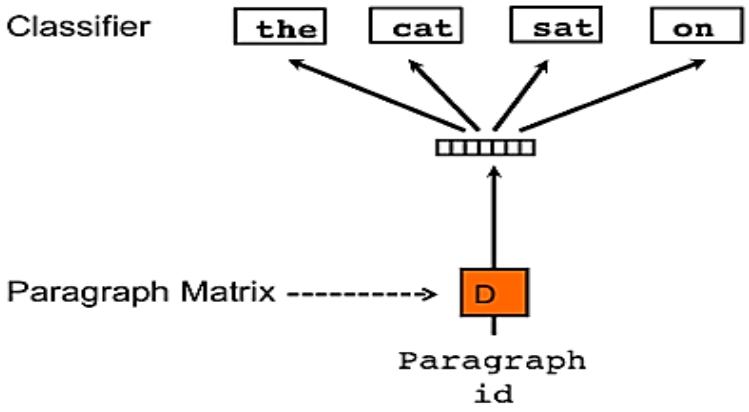


Figure 2.12 Distributed Bag of Words (DBOW) Doc2Vec

The training process updates word and document vectors iteratively to maximize the likelihood of predicting correct words in context. Techniques like negative sampling or hierarchical softmax are employed to accelerate training, similar to Word2Vec.

2.10 Dimensionality Reduction in NLP Tasks

Dimensionality reduction is a technique utilized in natural language processing (NLP) tasks to reduce the number of features (dimensions) in a high-dimensional space while preserving the essential information. NLP tasks often involve dealing with large feature spaces due to the vast vocabulary of words and the need to represent text data as numerical features for machine learning algorithms. By employing dimensionality reduction, several challenges related with high-dimensional data, such as computational complexity, overfitting, and the curse of dimensionality, can be addressed.

Feature selection methods, on the other hand, purpose to identify a subset of the most relevant features from the original feature space. The concept revolves around retaining only those features that meaningfully contribute to the specific task while eliminating irrelevant or redundant ones. In NLP, some popular feature selection methods include:

Document Frequency Thresholding: This technique involves eliminating words that appear in either too few or too many documents, as such words may lack significant discriminative power.

Mutual Information: This method measures the dependency between features and the target variable in the choice of informative features.

Chi-Square Test: By Evaluating the independence between features and the target variable, this test rejects non-informative features.

Feature Selection: Feature selection is beneficial when interpretability and model simplicity are essential or when working with limited computational resources.

The advantages of dimensionality reduction in NLP tasks include:

Improved computational efficiency: Reducing the dimensionality of the feature space results in faster training and prediction times for machine learning models.

Reduced risk of overfitting: High-dimensional feature spaces can cause models to overfit to the training data, while dimensionality reduction helps mitigate this issue.

Better generalization: Dimensionality reduction can lead to a more concise and informative representation of the data, maybe enhancing the generalization performance of machine learning models.Nevertheless, it's essential to be cautious when applying dimensionality reduction techniques, as reducing dimensionality too much can lead to the loss of significant information and theoretically degrade model performance. Careful evaluation and experimentation are necessary to determine the optimal level of dimensionality reduction for a specific NLP task.

2.11 Some Previous Research on Fake News Detection

The system developed an accurate and automated model to detect and forecast fake news by effectively capturing, evaluating, and integrating text, reactions, and sources. This research attempts to improve the accuracy and efficacy of the method in determining whether or not an article is fake. This paper has extensively discussed three unique manual modules for detecting fake news and acknowledges their limitations. The article's 'Text' is applied in the first module to determine whether the headings fit the content. This includes utilizing simple machine learning algorithms and natural language processing to extract textual properties and classify them as either true or fake. Nevertheless, the first module can result in false-positive circumstances when linguistic features are not taken into account. The second module dealt with studying the reaction to fake news. The comments and arguments of users against the news are utilized to determine the user's reaction to it.

Incendiary rhetoric is often utilized in the comments section of blatantly fake news. Social media is a great platform for understanding user sentiment about news, and simple classifiers can be used to determine whether it is fake or not. However, this approach proves to be time-consuming and labor-intensive. The third module is to determine the source of the news articles. The process entails examining the URL, researching the publication, and calculating the post score. Nevertheless, these approaches rely heavily on hand-crafted feature selection for classification, which poses a fundamental constraint. To address this limitation, the author has presented a Deep neural network CSI model capable of automatically selecting features, executing classification, and providing the decision of whether the news is fake or true [70].

The system proposed applying machine learning and network analysis techniques to determine the validity of a document. Their research investigated into the examination of different deception assessment approaches and their outcomes with the objective of devising a hybrid approach. The study places significant focus on two primary approaches, namely the linguistic and network approaches. The linguistic method involves training a machine learning algorithm on the text to categorize it as either false or real, relying on textual features and language patterns. The linguistic method includes training a machine learning algorithm on the text to categorize it as either false or real, relying on textual features and language patterns. On the contrary, the network method contains studying metadata and queries. Within the linguistic approach, some popular techniques are employed, such as identifying n-grams and grouping the words, utilizing probability context-free grammars to categorize based on rules and rhetorical structure, and discourse analysis. Social network behavior is frequently used to classify fake news articles. Finally, the author presents a strategy that integrates a highly sophisticated model capable of performing classification on linguistic features using multiple layers to achieve optimal performance. Furthermore, to achieve the best performance, both linguistic and network approaches must be combined and utilized in tandem [71].

The system demonstrated a fundamental approach for detecting fake news using the Navie Bayes algorithm. They illustrated the use of the Spam filtering method as a means to identify fake news. The data from the Facebook API is utilized to develop the model,

and additional labeled data is used for testing. Remarkably, even with a simple classification technique, the model achieved an accuracy of 76% [72].

The system used technology to deal with fake news and other internet phenomena by distinguishing related headlines from unrelated ones and further categorizing interconnected headlines. In their study, they investigated how headlines determine the perspective of the news content. To determine the authenticity of news stories, a rigorous methodology that includes lemmatization and n-gram classification was used. The model was developed using a set of classification approaches known as fine-grained classifiers. Remarkably, by using this technique, they achieved a weighted accuracy of 89% in determining whether news stories are fake or genuine [73].

2.12 Research Gaps

Based on an analysis of the existing literature conducted by other authors, good accuracy results were obtained in the models that were used. The system [72] employed the Naive Bayes classifier on a relatively small dataset including only 2000 occurrences, and they achieved a classification accuracy of 75%. However, it was noted that even better results could have been obtained had they used a larger dataset instead.

On an unlabeled dataset, a novel semi-supervised multitask learning approach based on Laplacian regularized logistic regression (SMTL-LLR) achieved an accuracy of 87% .However, it was observed that while the model performed well on unlabeled data, combining more unlabeled data could lead to an increase in noise and a subsequent decline in the performance of SMTL-LLR.

The Stochastic Gradient Descent classification algorithm was employed together with the TF-IDF text representation technique, achieving an accuracy of 77.2%. On the other hand, Ahmed, the system used a uni-gram TF-IDF as the text representation technique along with an LSVM classifier model, and they achieved a higher accuracy of 92% [74].

In the present literature studies, all the researchers employed the TF-IDF method for text feature extraction. A TF-IDF value or score indicate the relative importance of a term in the document based on its frequency of occurrence compared to the total number of words in that document, inverted by the total number of documents in the corpus that

contain this word. Nevertheless, it was observed that word embeddings and sentence embeddings were never employed to complete the text feature extraction technique.

Embedding refers to the method of encoding words and documents using a dense vector representation with n-dimensions that can be defined. Nevertheless, none of the previous studies incorporated probabilistic sentiment scores for fake news detection. Probabilistic sentiment scores are generated by a sentiment analysis approach that effectively classifies positive and negative sentiments. In our scenario, positive and negative sentiments in the news are methods of being true and fake. The existing sentiment model generated probabilistic scores by means of TF-IDF features with a high dimension but did not use dimension reduction techniques.

This study conducts a comparison and present the results of two models: the sentence embeddings-based SVM model and Naive Bayes with PSS against those models lacking PSS. To achieve these results, the system integrates Support Vector Machines and Naive Bayes classifiers will be integrated with a probabilistic sentiment model. Furthermore, the research will also include a comparison of evaluation results between the PSS model with dimension reduction and the PSS model without dimension reduction.

CHAPTER 3
BACKGROUND THEORY

This section provides an analysis of the foundational principles that form the basis of the proposed system. The PSS model, which is founded on the theory of information gain and entails feature selection from TFIDF features, is utilised by the proposed system to compute the probabilistic sentiment score. Furthermore, sentence embedding features are incorporated into our analysis by utilising an InferSent model.

3.1 Preprocessing

Data preprocessing[75] is a crucial step in preparing data, which entails applying several procedures to raw data in order to make it applicable to further data processing activities. Historically, it has served as a crucial component in the earliest stage of the data mining process. Recently, data preparation methods have been adapted to meet the training of machine learning models and AI algorithms, as well as for making conclusions using them. Data preprocessing aims to transform data into a format that is more understandable and efficient for tasks such as data mining, machine learning, and other data science pursuits. These methods are often used in the first stages of machine learning and AI development pipelines to ensure the production of accurate results. Algorithms that learn from data may be described as statistical equations that manipulate values obtained from a database. According to the well-known phrase, "if garbage goes in, garbage comes out". The success of your data project relies on the prerequisite of inputting high-quality data into the computers. Data obtained from real-world situations always include both noise and missing values. This occurs as a result of human mistakes, unexpected events, technological difficulties, or a range of other challenges. Algorithms are often poorly equipped to manage incomplete and noisy data due to their failure to handle missing values and the disruption caused by noise in the sample's real pattern. Data preparation seeks to address these issues by comprehensive manipulation of the available data. Data preprocessing primarily involves assessing the quality of the data.

This quality assessment encompasses the following aspects:

Accuracy: Verifying the correctness of the entered data.

Completeness: Ensuring that all necessary data is available and recorded.

Consistency: Confirming that the data remains uniform across all instances and discrepancies are resolved.

Timeliness: Ensuring that the data is regularly updated and remains current.

Believability: Ensuring the trustworthiness and reliability of the data.

Interpretability: Assessing the comprehensibility and clarity of the data for understanding.

3.2 Major Tasks in Data Preprocessing

There are 4 major tasks in data preprocessing – Data cleaning, Data integration, Data reduction, and Data transformation.

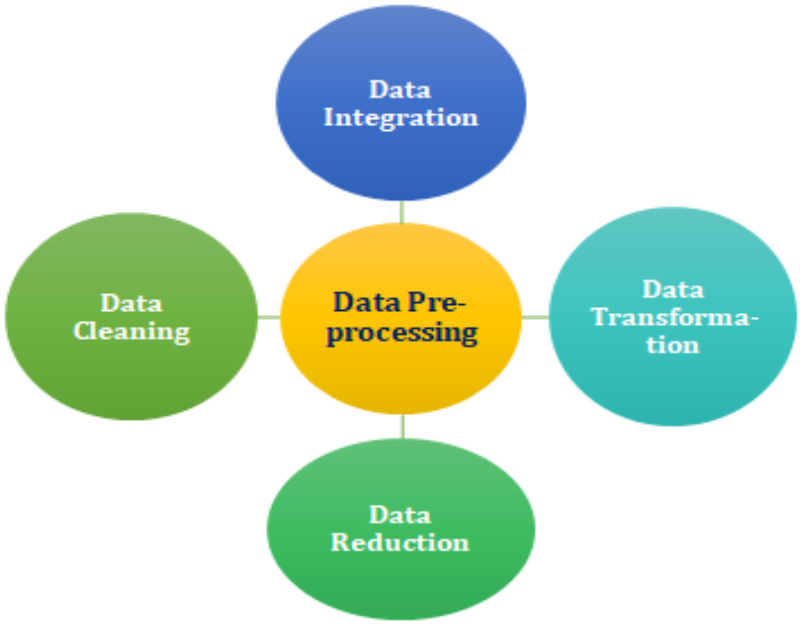


Figure 3.1 Data Preprocessing Pipeline

3.2.1 Data cleaning

Text data cleaning[76] involves preparing textual data for analysis by removing irrelevant or noisy information and standardizing its format. Common steps include removing HTML tags and non-textual characters, lowercasing, removing punctuation and stopwords, stemming or lemmatization, spell checking, tokenization, removing rare words, handling contractions and abbreviations, normalization, handling URLs, and managing

emoji and emoticons. These steps ensure that the data is consistent and usable for further analysis or modeling, with variations based on data nature and analysis requirements.

3.2.2 Data Integration

Data integration[77] involves merging multiple sources into a unified dataset, a critical aspect of data management. Key considerations include schema integration, which aligns metadata from diverse sources, entity identification to recognize shared entities across databases, and detecting and resolving discrepancies in data values such as differing attribute formats. Examples include varying date formats like "MM/DD/YYYY" or "DD/MM/YYYY."

3.2.3 Data Reduction

Data Reduction is a process aimed at decreasing data volume, facilitating easier analysis while maintaining comparable results. This reduction also contributes to conserving storage space. Some techniques for data reduction include dimensionality reduction, numerosity reduction, and data compression. Dimensionality reduction is essential for real-world applications dealing with large datasets. It involves reducing random variables or attributes to decrease dataset dimensionality while retaining original characteristics. This consolidation of attributes aids in reducing storage requirements and computation time. However, highly dimensional data may encounter issues known as the "Curse of Dimensionality."Numerosity Reduction aims to decrease data representation volume without any loss of data.Data compression refers to transforming data into a compressed format, which can be either lossless or lossy. Lossless compression retains all information, while lossy compression reduces information by eliminating only unnecessary components.

3.2.4 Data Transformation

Data Transformation refers to altering the format or structure of data. This step can vary in complexity depending on specific needs. Various methods are employed for data transformation. Smoothing involves using algorithms to eliminate noise from the dataset, enabling identification of crucial features for prediction. Even simple alterations detected

through smoothing can aid in prediction accuracy.Aggregation entails summarizing and presenting data from multiple sources into a unified form for analysis. This step is crucial as data accuracy hinges on both quantity and quality. Higher quality and quantity of data yield more relevant results.Discretization involves partitioning continuous data into intervals, reducing data size. For instance, instead of specifying exact class times, intervals such as (3 pm-5 pm) or (6 pm-8 pm) are utilized. Normalization scales data to a smaller range, typically from -1.0 to 1.0, for representation.

3.3 Logistic Regression

Logistic Regression (LR), within the realm of supervised machine learning, operates by leveraging independent variables to predict the dependent variable, thereby functioning as a predictive algorithm. The fundamental objective of Logistic Regression revolves around establishing the relationship between the independent variables, commonly referred to as features, and the dependent variables targeted for prediction .Logistic Regression is predominantly utilized to forecast the outcome of a categorical dependent variable, often expressed in a categorical or discrete manner. These categorical representations can manifest as binary choices such as "Yes" or "No," "0" or "1," or "true" or "false," devoid of precise numerical delineations. The methodology of logistic regression relies on the sigmoid function, characterized by its distinctive "S"-shaped curve, which facilitates the prediction of values constrained between 0 and 1, diverging from the linear regression paradigm. This sigmoid function facilitates a probabilistic evaluation of outcomes by mapping real numbers to a bounded interval. Figure 3.2 illustrates the visualization of LR.

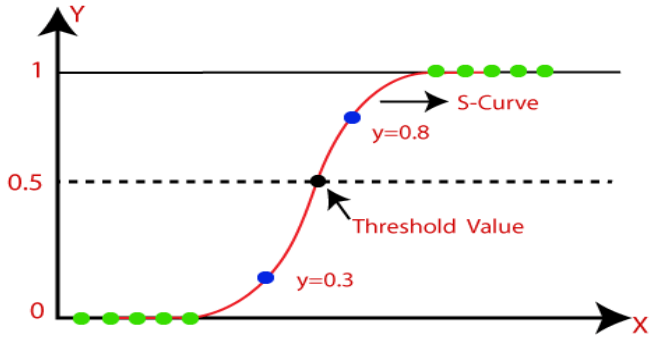


Figure 3.2 Logistic Regression

The input features in this specific scenario concern characteristics of a news article, whereas the outcome variable represents the truth or fake of the article. The input features are supposed to be denoted as X, with X equal to $[x_1, x_2, \dots x_n]$, and the outcome variable is to be denoted as y, with y taking on binary values of either 0 (which means a real news article) or 1 (signifying a fake news article). Using the input features X, the logistic regression model attempts to forecast the probability that an article (y = 1) is fake. The probability that an article has been faked is computed by this model using the input characteristics of the article. The process utilizes a logistic or sigmoid function to convert the linear combination of the input features into a binary value between 0 and 1.There are three primary types of logistic regression:

Binary logistic regression: This type is employed to forecast the likelihood of a binary outcome, like yes or no, true or false, or 0 or 1. For instance, it could predict whether a customer will churn, if a patient has a disease, or if a loan will be repaid.

Multinomial logistic regression: This form predicts the probability of one out of three or more potential outcomes, such as the type of product a customer might purchase, the rating they might give a product, or the political party they might vote for.

Ordinal logistic regression: This method predicts the probability of an outcome that follows a pre-established order, like the degree of customer satisfaction, the severity of an illness, or the stage of cancer.

Equation 3.1 defines the logistic function, which maps the linear combination of the input features and the z-coefficients that represent them.

$$sigmoid(z) = \frac{1}{1+e^{-z}} \qquad \text{Equation (3.1)}$$

The mathematical expression used to calculate z is Equation 3.2, which contains the weights or coefficients that are linked to each input feature.

$$z = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n \qquad \text{Equation (3.2)}$$

The model obtains these coefficients during the training phase, where it makes the necessary adjustments to them in order to minimize the loss function. After the coefficients have been determined the model is capable of predicting the outcome variable for newly

added articles through an evaluation of the likelihood that it will equal 1 using the input features. The sigmoid function is utilized to perform this computation by converting the value of z, which is obtained by combining the coefficients of the input features, into the predicted probability. The probability denoted by Equation(3.3) represents the level of confidence that the model has in classifying the article as fake.

$$P(y = 1|X) = sigmoid(b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n) \quad \text{Equation (3.3)}$$

Through a comparison of the estimated probability with a set threshold of 0.5, the model discerns the truth or falsity of the article in this particular context.

3.4 Cost Function in Logistic Regression

In logistic regression, the cost function differs from linear regression. Instead of using the Mean Squared Error, which calculates the difference between the predicted and actual values of y, logistic regression employs a cost function derived from the maximum likelihood estimator. The graphical representation of the cost function in logistic regression varies from that of linear regression.

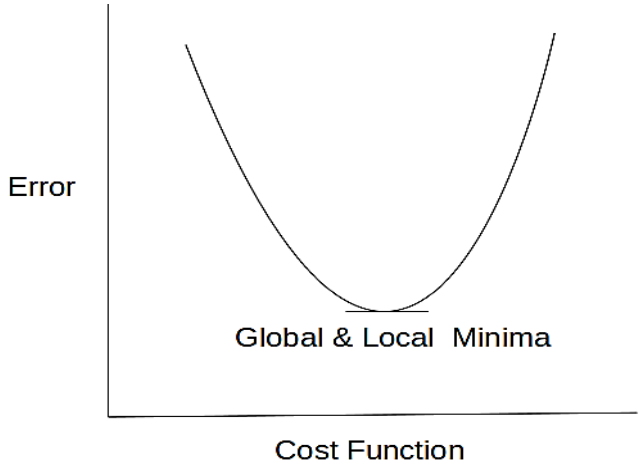


Figure 3.3 Linear Regression Cost Function

$$J = \frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n} \quad \text{Equation (3.4)}$$

When applying logistic regression, Yi represents a non-linear function ($\hat{Y}$=1/1+ e-z). Utilizing this function in the Mean Squared Error equation mentioned above results in a non-convex graph with multiple local minima, as illustrated.

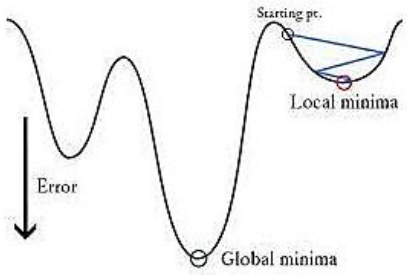


Figure 3.4 Non-convex Graph with Many Local Minima

The issue lies in the fact that this cost function produces outcomes with local minima, posing a significant problem as it could lead to overlooking the global minimum and increasing our error. To address this challenge, an alternative cost function is used for logistic regression known as log loss, which is also derived from the maximum likelihood estimation technique.

$$Log\ loss\ = \frac{1}{N}\sum_{i=1}^{N} -(y_i * log(\widehat{Y}_i) + (1 - y_i) * log(1 - \widehat{Y}_i)) \qquad \text{Equation (3.5)}$$

3.5 Information Gain

Information gain[78] (Info-gain) can also be applied to the selection of features, by evaluating the gain of each variable in the context of the target variable. The method described by for feature selection involved the integration of information gain parameters to improve the accuracy of classification algorithms. Mutual information between the two random variables is calculated in Info-gain. Mutual information I (X: Y) is the amount of uncertainty in X due to the knowledge of Y. Mathematically, information gain is defined as shown in "Equation (3.6)".

$$I(X:Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \qquad \text{Equation (3.6)}$$

where,

p (x, y) means the joint probability function of X and Y,

p(x) is the marginal probability distribution function of X and

p(y) is the marginal probability distribution function of Y.

In machine learning and data analysis, information gain is a statistical metric utilised to determine the significance of a feature in predicting a target variable. It quantifies the extent to which a specific attribute mitigates uncertainty in forecasting the final result. Information gain, within the framework of feature reduction, enables the identification of the most informative features while excluding those that are less significant, thereby decreasing the dimensionality of the dataset.The procedure entails determining the information gain for each feature by evaluating its capability to differentiate between various classes of the target variable. Features that possess a greater information gain are deemed more crucial for the purpose of classification and are therefore retained. Conversely, features that have a lower information gain are eliminated.

Through the utilisation of information gain for feature reduction, the dimensionality of the dataset gets reduced, thereby facilitating the processing of the data by machine learning algorithms and potentially enhancing the predictive model's performance.

3.6 Probabilistic Sentiment Score

The proposed system has implemented a probabilistic sentiment model[79] as depicted in Figure 3.5 that can classify positive and negative effectively. To implement the PSS model, the system first perform pre-processed text, and it then counts the frequency of word to compute the TF-IDF score. The TF-IDF values extracted by the system are in a considerable amount of over 20,000 words. Therefore, the proposed system uses information gain theory to extract the best attributes and reduce the features dimensions. The system then applies logistic regression methods based on the selected feature set to get the PSS values. The system employed TF-IDF method to vectorize the pre-processed data. Using the TF-IDF vectorizer, one can tokenize documents, build the vocabulary, calculate inverse document frequency weights, and process incoming document effectively. TF-IDF computes the relevance of a term in a incoming documents by examining its importance in the context of a document.

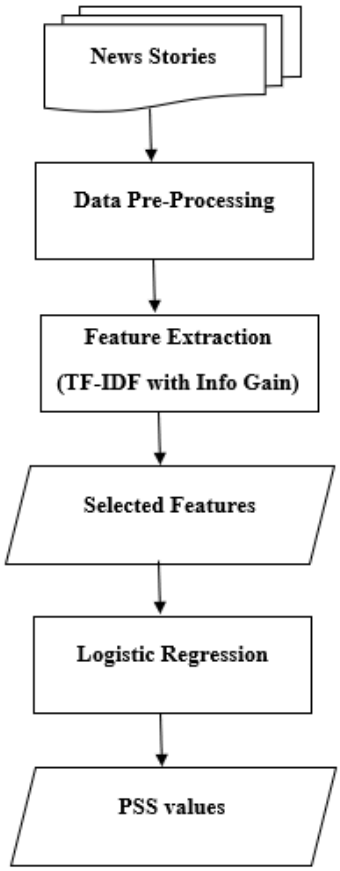


Figure 3.5 The System Flow of PSS Model

According to "Equation (3.7)", TF value is derived by dividing the occurrence of a specific term in a document by the total number of terms in that document. IDF score is obtained by utilizing "Equation (3.8)". The TF-IDF score for a term is obtained by multiplying TF and IDF scores, as demonstrated in "Equation (3.9)".

$$TF = \frac{no.\ of\ \ times\ the\ term\ appears \text{ in } the\ document}{total\ no.\ of\ \ terms \text{ in } the\ document} \qquad \text{Equation (3.7)}$$

$$IDF = \log \frac{no.\ of\ \ documents\ in\ the\ corpus}{no.\ of\ \ documents\ in\ the\ corpus\ contain \text{ in } the\ term} \qquad \text{Equation (3.8)}$$

$$TF\_IDF = TF * IDF \qquad \text{Equation (3.9)}$$

The utilization of information gain theory for feature selection is a pivotal step in enhancing the effectiveness of the sentiment analysis model. By evaluating the gain of each variable in relation to the target variable, this technique aids in identifying the most informative features, thereby reducing the dimensionality of the feature space and improving the model's accuracy.

After selecting the important features from the TF-IDF scores using information gain for all terms in the document, the proposed system employs the Logistic Regression classifier to derive probabilistic sentiment scores (PSS). Logistic Regression is a supervised learning algorithm used for binary classification tasks like sentiment analysis. It learns a mapping from input features (in this case, the TF-IDF scores of selected terms) to a binary output (positive or negative sentiment). The output of the Logistic Regression model can be interpreted as the probability of a document belonging to a particular sentiment class.

By combining the TF-IDF features with the Logistic Regression classifier, the proposed technique aims to effectively classify documents as either positive (real) or negative (fake), thereby facilitating fake news detection. The use of TF-IDF for feature extraction helps capture the importance of terms within the context of the entire document set, while Logistic Regression provides a probabilistic approach to sentiment analysis, allowing for nuanced classification decisions.

3.7 Sentence Embedding Text Representation Technique for Text Classification

Machine learning algorithms operate primarily on numerical data. Converting textual information into a numeric representation is termed vectorization. In the realm of natural language processing (NLP), word embedding constitutes the conversion of words into dense, continuous vectors within a multi-dimensional space. However, word embeddings may not encapsulate the entirety of word semantics, especially for words with various interpretations or ambiguity.

In contrast to word embeddings, sentence embeddings provide a means to convert complete sentences into vectorized representations, rather than focusing solely on individual words. These methodologies strive to transform word sequences of varying lengths into fixed-length representations, offering utility across diverse NLP tasks. Noteworthy applications of such representations include text classification, sentiment

analysis, information retrieval, machine translation, and question-answering systems. In recent years, prominent sentence encoders like Google's BERT and USE, Facebook's InferSent, and AllenAI's SciBERT and ELMo have gained substantial traction within the NLP community.

InferSent employs a deep neural network architecture, typically based on recurrent neural networks (RNNs) or transformers, to encode the input sentences into fixed-length vectors. These vectors serve as embeddings that encapsulate the semantic meaning of the sentences. The trained model can then be used to generate embeddings for new sentences, which can be employed in various natural language processing tasks such as sentiment analysis, text classification, and information retrieval [80].

The utility of sentence embeddings lies in their capacity to encapsulate both semantic meaning and contextual nuances, thereby facilitating a diverse array of downstream tasks in natural language processing. These tasks span from text classification and semantic similarity assessment to clustering and more. The architecture of InferSent is structured around two primary modules: a sentence encoder and an NLI (Natural Language Inference) classifier. Within this framework, the sentence encoder plays a pivotal role in converting input sentences into fixed-length vectors, effectively capturing their underlying meaning and contextual intricacies. Subsequently, the NLI classifier leverages these encoded vectors to facilitate text classification endeavors. An illustrative depiction of the core workflow of InferSent can be observed in Figure 3.6.

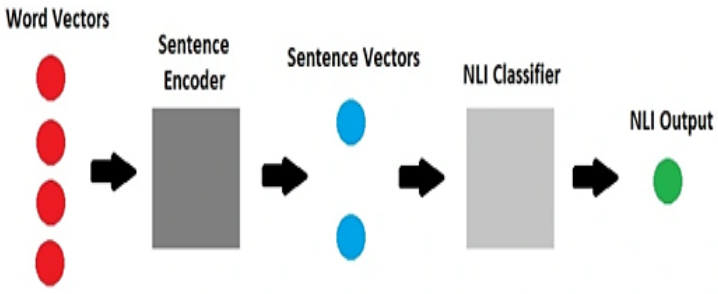


Figure 3.6 General Flow of InferSent

3.8 Machine Learning Classifiers

Fake news categorization involves automatically identifying if a certain piece of news or information is real or fake[81]. It usually entails using machine learning algorithms to examine multiple features of news, such as language, source, and context, to forecast its reliability. The technique entails training a classification model using a dataset of labelled news items, with each article categorized as either real or fake. The program extracts patterns from the data to categorize current news stories into one of two groups. Fake news classification algorithms may differ in complexity and accuracy based on aspects including data quality, classification features, and machine learning algorithm selection. Popular algorithms for identifying fake news include Naive Bayes, Logistic Regression, Support Vector Machines, Random Forest, and Neural Networks. Fake news classification algorithms are evaluated using measures including accuracy, precision, recall, and F1-score to assess their ability to properly identify real and fake news items. Fake news categorization is essential in addressing the spread of misinformation and disinformation online by automatically detecting and marking potentially misleading material. The general pipeline of fake news classification is illustrated in the following figure.

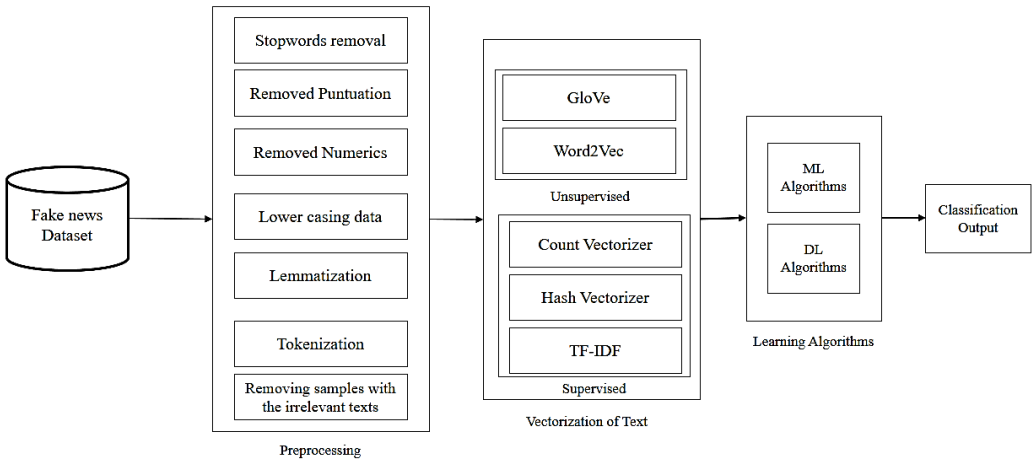


Figure 3.7 General Pipeline of Fake News Classification

3.8.1 Naïve Bayes

The Naïve Bayes algorithm, a cornerstone of supervised learning, draws its foundation from Bayes' theorem and is instrumental in addressing classification problems across various domains. Particularly prevalent in text classification tasks involving high-

dimensional datasets, Naive Bayes[82] serves as a reliable and efficient tool for constructing machine learning models capable of rapid predictions. As one of the most straightforward and effective classification algorithms, the Naive Bayes classifier facilitates the swift development of predictive models. Operating as a probabilistic classifier, it bases its predictions on the likelihood of an object's occurrence within a given class. This characteristic makes it invaluable in applications such as spam filtration, sentiment analysis, and categorization tasks.

Bayes' Theorem, the theoretical underpinning of the Naive Bayes algorithm, enables the calculation of the probability of an event's occurrence given the probability of another event that has already transpired. Expressed mathematically, Bayes' Theorem is encapsulated by the formula:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \qquad \text{Equation (3.10)}$$

where,

A, B = Events

P(A | B) = Probability of A given B is true

P(B | A) = Probability of B given A is true

P(A), P(B) = The independent probabilities of A and B

This formula serves as the cornerstone for probabilistic reasoning and inference in the Naive Bayes framework, facilitating the algorithm's ability to make informed predictions based on available data. Through its probabilistic nature and reliance on Bayes' Theorem, the Naive Bayes algorithm emerges as a versatile and powerful tool for classification tasks, offering rapid insights and predictions across diverse domains.

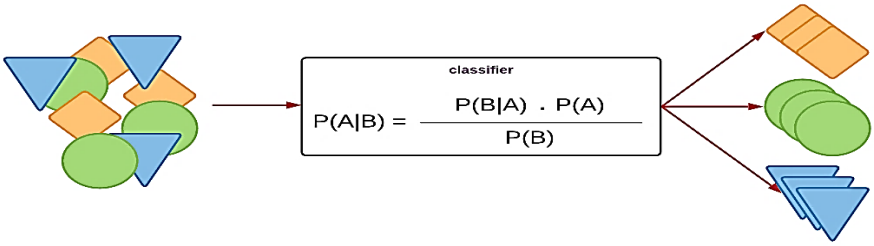


Figure 3.8 Naïve Bayes Classification

3.8.2 Support Vector Machine

SVM is one of the top classification machine learning algorithms. It may be used on sets with only two classes for classification. By dividing the sets, the multi-class dataset classification problem can be converted to a binary problem. Because training requires fewer sets, the partitioning problem is minor. The outcome is a binary SVM comparison. It works by constructing the best hyperplane that divides points of distinct classes. It creates a plane or a collection of planes in a large or multidimensional space. The plane with the greatest distance to the nearest training data performs better classification. The smaller the margin, the smaller the error. The purpose of SVM is to add the margins between two classes. There are many methods for working with regression and classification. SVM is interesting for its ability to work on problems with featured elements as well as difficulties that cannot be worked on. It is a simple approach used for dataset separation and feature extraction that acts on familiar sets and builds a decision plane to split the sets.

The consequences obtained by constructing the line with more distance to the closest training data.

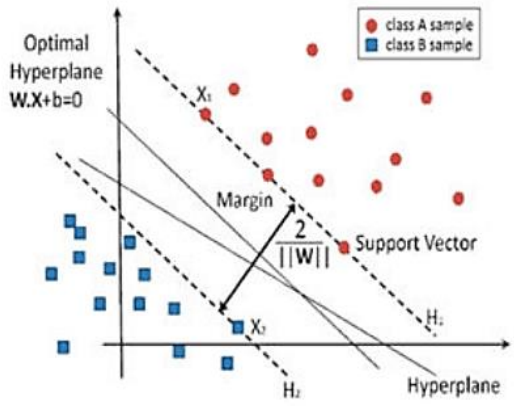


Figure 3.9 Support Vector Machine

The linear plane gives better results for binary classifications.SVMs are also used to classify stage of cancer. For any category, a binary SVM classifier is trained even if every document in training belongs to that category or no. any medical report may not contain a single stage. The SVMs work using the functions available in the toolkit. The machine is trained from huge list of results with different levels. After working on test data, it gives a score that acts as a threshold value to know if anything is updated that falls into

the same class. To increase the information, we need to train while still giving out meaningful outputs on the same data.

Given a data which has an input denoted as $\boldsymbol{xi} \in R\ d$ and output in the form of target denoted as $yi \in \{-1, +1\}$ for i = 1, 2, ..., n, where n is the amount of data. Then, it is possible to separate the class -1 and +1 using a hyperplane with dimension n presented in the following equation:

$$W^T.X_i + b = 0 \qquad \text{Equation (3.11)}$$

The hyperplane formed can separate the data into two classes with positive or negative values such that those in the positive class are labeled as

$$y_i = +1 \qquad \text{Equation (3.12)}$$

for i = 1, 2, ..., N and this means x_i can be defined as follows.

$$W^T X_i + b \geq 1 \qquad \text{Equation (3.13)}$$

Then, when the $\boldsymbol{xi}$ data belongs to the negative class where $y_i = -1$ for i = 1, 2, ..., N, it can be defined as:

$$W^T X_i + b \leq -1 \qquad \text{Equation (3.14)}$$

Therefore, for each x_i data with label $yi \in \{-1, +1\}$ for i = 1, 2, ..., N can be defined as:

$$y_i\ (W^T X_i + b) \geq 1 \qquad \text{Equation (3.15)}$$

We will go over the proposed for for classification fake news based on sentence structure in the next chapter.

3.9 Summary

This chapter explores the fundamental ideas that form the basis of our studies. A detailed examination of the foundational ideas is provided that underpin the work, clarifying their importance and relevance within the scope of our research.This chapter introduces the PSS (Probabilistic Sentiment Score) model, a new method for quantifying sentiment analysis using probabilistic calculations. The model is intricately designed utilising a logistic regression framework, providing a strong approach for calculating

sentiment scores from a probabilistic aspect.We also investigate the crucial process of feature selection, specifically focusing on the idea of knowledge acquisition. The significance of choosing important and informative characteristics will be explored to improve the precision and effectiveness of our sentiment analysis model. The chapter also discusses the implementation of sentence embedding using InferSent. This innovative method enables us to depict phrases in a multi-dimensional space, conveying their semantic content in a detailed and subtle way. It is to improve the depth and accuracy of the sentiment analysis algorithm by using InferSent and to provide in-depth analysis of the two classification methods used in the proposed model: Support Vector Machine (SVM) and Naive Bayes (NB). The classification algorithms are well-known for their efficacy in managing intricate datasets and are meticulously included into our model to enhance precise sentiment analysis.

This chapter provides a thorough examination of the theoretical underpinnings, approaches, and procedures used in our study. By clarifying these essential elements, the researcher provides the foundation for a more profound comprehension of the suggested sentiment analysis model and its significance in practical scenarios.

CHAPTER 4
FAKE NEWS DETECTION SYSTEM BASED ON PROBABILISTIC SENTIMENT SCORE AND SENTENCE EMBEDDING

The proposed system has used concatenated features probabilistic sentiment score and sentence embedding to enhance the fake news detection accuracy.The integration of probabilistic sentiment scores (PSS) and sentence embeddings is a technique that seeks to improve the efficacy and precision of identifying fake data, with the objective of enhancing the detection of fake news. Utilizing PSS, which calculates the probability that a news article is real or fake according to its sentiment, and sentence embeddings, which extract the semantic and contextual details from textual data, is required for that approach.By integrating these two methodologies, the detection system acquires an improved understanding of the complexities of language and emotion that are present in news articles, leading to a more accurate categorization. The sentiment polarity of the text is visible through the probabilistic sentiment scores, whereas the semantic meaning and syntactic structure of sentences are encoded via sentence embeddings.

By employing this integrated methodology, a thorough examination of news articles is possible, in which both the sentiment conveyed and the contextual information incorporated are considered. Through the implementation of these sophisticated methodologies, fake news detection systems are capable of more accurately differentiating authentic from misleading material, thus reducing the dissemination of false information and enhancing the general reliability of sources of news.

4.1 Datasets Description

The ISOT dataset[83], established by the Information Security and Object Technology (ISOT) Research Lab, stands as a pivotal resource in the realm of fake news detection. Table 2 provides a glimpse into the dataset's composition, comprising a diverse array of articles sourced from contemporary media coverage. Real news articles, meticulously curated from Reuter.com, juxtapose alongside their counterfeit counterparts,

gleaned from websites flagged as unreliable by esteemed fact-checking entities like PolitiFact and Wikipedia.

Spanning a broad spectrum of subjects, the dataset encompasses articles spanning a plethora of themes, although a notable emphasis is observed on political and global affairs. This rich diversity affords researchers a comprehensive panorama of the contemporary media landscape, facilitating nuanced analyses and insights into the proliferation of misinformation. The distribution of articles categorized as "Real" and "Fake" is visually depicted in Figure 4.1, underscoring the dataset's balanced representation and its potential for robust empirical investigations into fake news phenomena across various domains. The ISOT dataset comprises two distinct labels: real and fake news. Within the real news category, there are 21,417 samples, distributed across two columns: "politicsNews" containing 11,272 samples and "worldnews" containing 10,145 samples as depicted in Figure 4.2. These news articles were amassed during the years 2016 and 2017 as shown in Figure 4.3. Conversely, the fake news category encompasses 23,450 samples and is characterized by six columns: "News" with 9,050 samples, "Politics" with 6,833 samples, "Left News" with 4,454 samples, "Government News" with 1,568 samples, "US_News" with 775 samples, and "Middle-East" with 770 samples, as depicted in Table 4.1. Overall, the combined real and fake news data exhibit eight columns and a total of 44,867 samples.

After cleaning and processing the data, a total of 44,658 samples were acquired. This dataset was then split into 40,192 samples for training and 4,466 samples for testing purposes.The ISOT dataset information is summarised in a Table 4.1, Figure 4.1, Figure 4.2 and Figure 4.3.

Table 4.1 News Categories and Number of News Articles per Category

Type	Number of Articles	Subjects	Number of News
Real	21417	World News	10145
		Political News	11271
Fake	23481	Government News	1570
		Middle East	778
		US News	783
		Left-News	4459
		Politics	6841
		News	9050

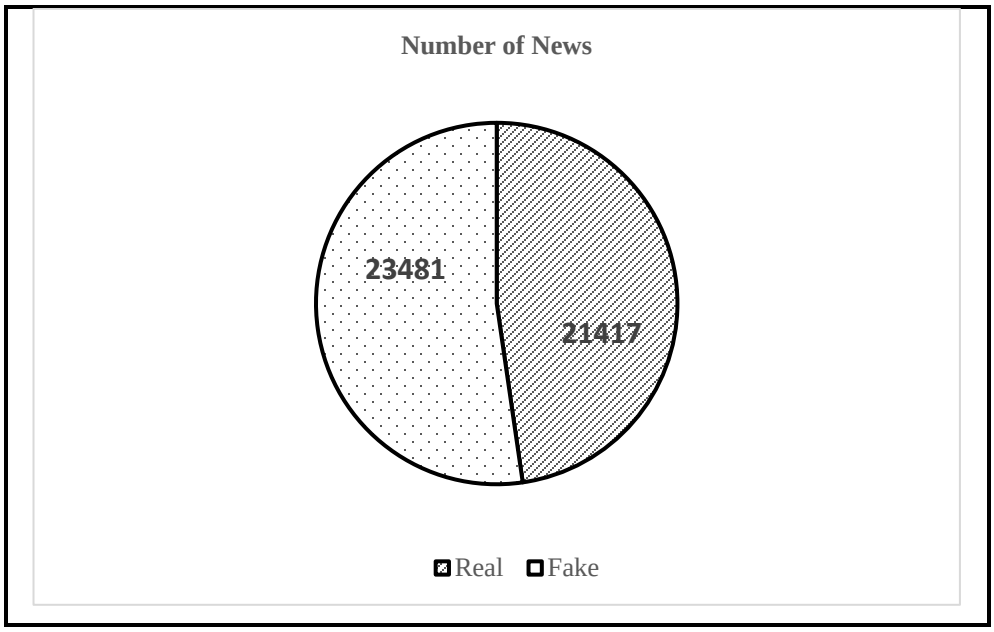


Figure 4.1 Real and Fake News Count of ISOT Dataset

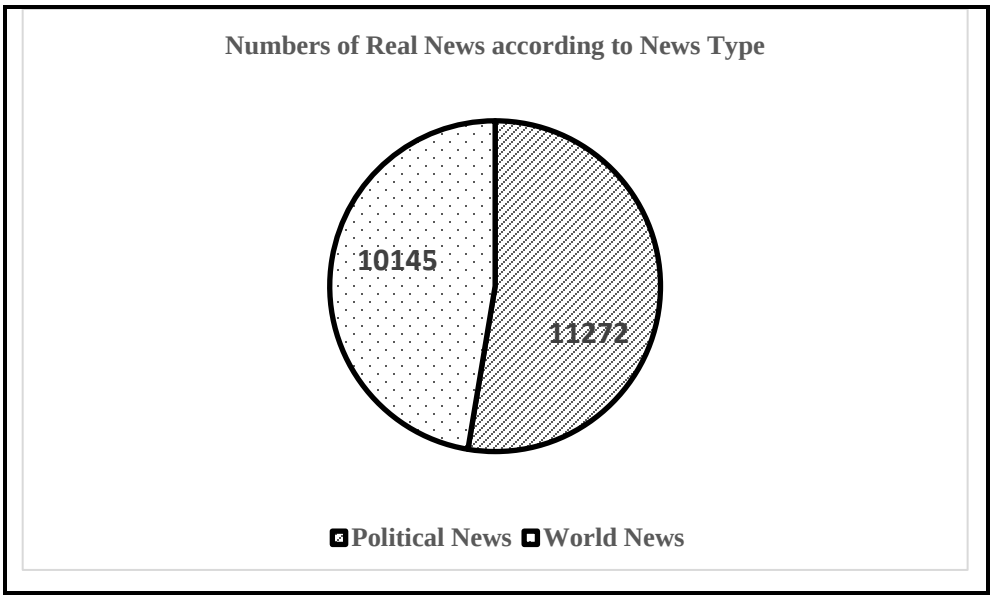


Figure 4.2 Number of Real News according to News Type of ISOT Dataset

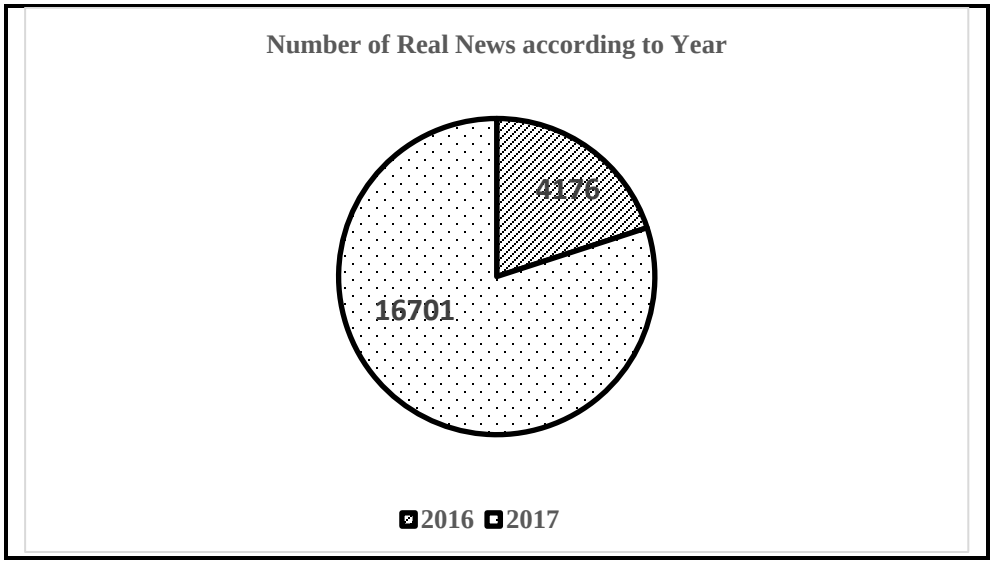


Figure 4.3 Number of Real News according to Year of ISOT Dataset

Furthermore, the proposed system also implements fake news detection using other fake news dataset from Kaggle to highlight the contribution points. The dataset contains 6206 news articles meticulously curated to represent a balanced distribution between fake and real news labels . This equilibrium ensures a robust foundation for analysis and model training across various applications in the domain of natural language processing and media

studies. To access and download the dataset through the provided link [84] . This repository serves as a valuable resource for researchers, educators, and enthusiasts alike, fostering exploration, experimentation, and advancement within the realm of news authenticity detection and related field. The dataset was then divided into 80% for training and 20% for testing purposes. The nature of the dataset is depicted in the following Figure 4.4.It contains 3092 fake articles and 3114 real news articles.

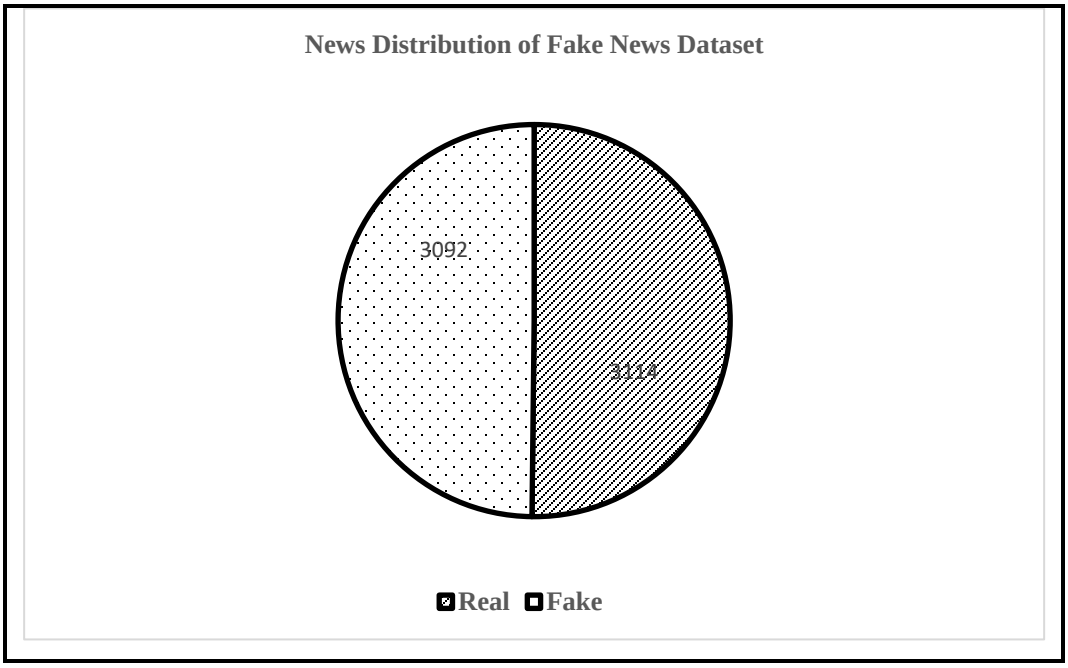


Figure 4.4 News Distribution of Fake News Dataset

4.2 The Proposed System Design

The methodology proposed for the detection of fake news is a comprehensive approach that integrates various techniques, including a Probabilistic Sentiment Score (PSS) model, sentence embeddings, and feature selection processes. At the core of the methodology lies the fusion of these components to ensure robust and accurate detection of misinformation. Preprocessing the data is begun to ensure its cleanliness and readiness for analysis. This step involves tasks such as handling missing values, removing duplicates, and standardizing the text format.

The calculation of Probabilistic Sentiment Scores (PSS) is a pivotal aspect of our methodology. By employing techniques such as TF-IDF, information gain, and logistic

regression (LR), we construct a model capable of assigning sentiment scores to news articles, thereby providing insights into their potential credibility. To enhance the representation of textual information, sentence embeddings are incorporated into our methodology. This allows us to capture both contextual information and word semantics effectively, thereby enriching the feature space and improving the performance of the model.

The proposed system consists of five core elements:

1. Data preprocessing
2. Calculation of Probabilistic Sentiment Scores (PSS)
3. Sentence embedding
4. Feature Combination and Concatenation
5. Classification using Naive Bayes and Support Vector Machine algorithms

Each of these elements plays a crucial role in the overall process of detecting fake news. The high-level perspective of the proposed system's process is depicted in Figure 4.5, providing a comprehensive overview of the workflow. Detailed explanations of each component are elaborated upon in subsequent sections, offering insights into the underlying methodologies and techniques employed.

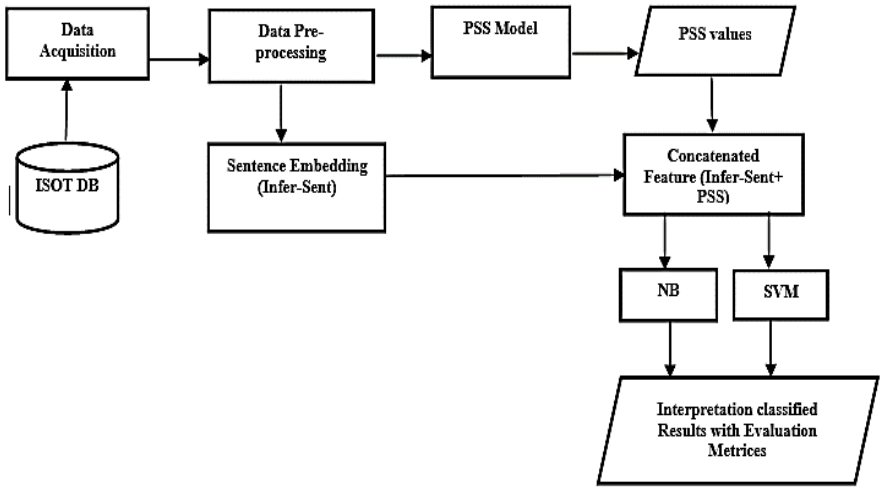


Figure 4.5 Overall Proposed System Design

4.3 Data Pre-processing

In light of the fact that news items obtained from social media platforms are often unstructured, extensive, and prone to noise, the stage of data pre-processing plays a crucially important role. Pre-processing procedures guarantee the dependability and uniformity of the data inputted into machine learning models, thus augmenting the efficacy of subsequent analyses. Text analysis frequently utilises methods including feature extraction, POS (part of speech) labelling, tokenization, stemming, and stop word removal to prepare the data for subsequent processing . The four fundamental stages encompassing data pre-processing, as implemented in the present research undertaking, are illustrated in Figure 4.6. Initially, duplicate and missing value rows are eliminated. Subsequently, special characters, punctuations, numbers, alphanumeric text, non-English words, and stopwords are removed. Tokenization is then applied to split sentences into tokens, followed by transformation into lemma form through lemmatization. In order to provide a firm basis for later analysis and model building, these processes help to organise, clean, and arrange the raw textual data from whence it was obtained.

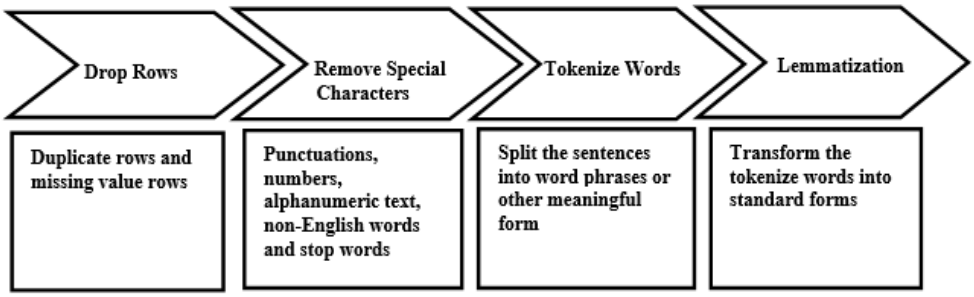


Figure 4.6 Steps of Data Preprocessing

4.4 Calculation of Probabilistic Sentiment Scores (PSS)

Compute the probabilistic sentiment score for each entry in the dataset, we employ the aforementioned approach. The PSS model effectively discerns between positive (real) and negative (fake) sentiments with a high degree of accuracy. Figure 4.7 illustrates the scores assigned by the PSS model to ten news articles.

From these two columns, the first one represents negative sentiment, while the second column represents positive sentiment. The sum of the values in both columns equals 1,

indicating that each sentiment category is represented proportionally. Therefore, in these two columns, the sentiment is represented by the column with the larger value.

	0	1
0	0.972079	0.027921
1	0.977765	0.022235
2	0.976047	0.023953
3	0.607438	0.392562
4	0.994298	0.005702
5	0.929328	0.070672
6	0.086406	0.913594
7	0.169626	0.830374
8	0.985901	0.014099
9	0.960284	0.039716

Figure 4.7 Probabilistic Sentiment Scores of 10 News Articles

A Probabilistic Sentiment Score (PSS) in negative sentiment column, it equals to or greater than 0.5 signifies that the news is fake, whereas a score below 0.5 suggests it is real. For instance, we classify news articles with IDs "0" and "1" as "Fake," and articles with ID "6" as "Real," and so on. Consequently, we select only one column from these two, depending on which one has the higher value, to determine the predominant sentiment. In order to integrate these scores with the set of sentence embedding features, one column of PSS values is removed. Table 4.2 displays the PSS values for sample news articles. Based on these values, we classify news articles with ID "0" and "1" as "Fake," and article with ID "6" as "Real," and so forth.

Table 4.2 The PSS for Samples of 10 News Articles

News ID	PSS score
0	0.027921
1	0.022235
2	0.023953
3	0.392562
4	0.005702
5	0.070672
6	0.913594
7	0.830374
8	0.014099
9	0.039716

4.5 Sentence Embedding using InferSent

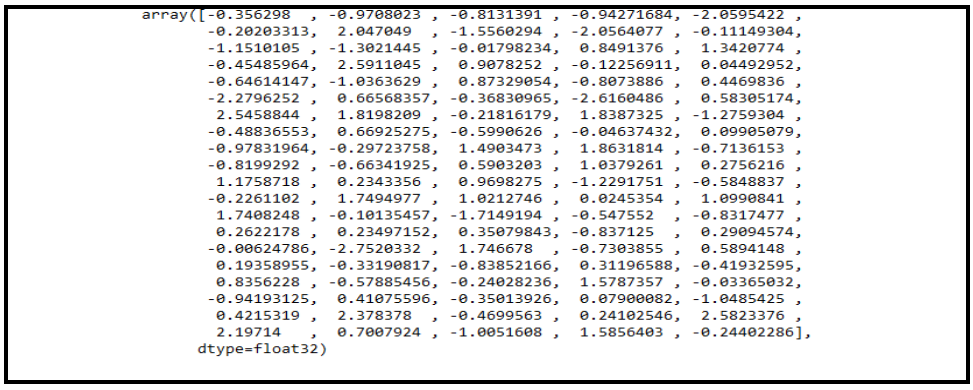
There are many sentence embedding methods for feature vector representation. Among them, InferSent is more suitable for large and complex rich data than other methods. Therfore, the proposed system uses InferSent sentence embedding methods for feature generation process. InferSent is a complex approach used in fake news detection to convert textual statements into concise numerical representations in a continuous vector space. This approach uses a pre-trained deep learning model designed to produce fixed-length vector embeddings for input texts. These embeddings act as concise representations that capture both the semantic content and contextual complexities of the sentences, making analysis and comparison more efficient.

When trying to detect deceptive or misleading information typical of false news, sentence embeddings created by InferSent are quite beneficial. InferSent enables machine learning models to effectively understand and categorise textual input by converting language patterns, syntactic structures, and semantic clues into compact numerical representations. Therefore, it helps to identify tiny differences and differentiate between

real and false news pieces. Using sentence embedding with InferSent greatly improves the analysis and understanding of text in fake news detection systems. This improvement helps get better and more accurate categorization results, therefore enhancing the efficiency of identifying fake news. The InferSent model is used to create sentence vectors by using the infer_vect technique from the gensim Doc2Vec model in this research. Parameter settings are methodically specified for the Doc2Vec model to guarantee optimum performance: vector_size=100, window=2, min_count=1, and epochs=100. Throughout the investigation, the default parameters of the InferSent model are adopted which are described as follows:

Table 4.3 The Parameter Value of Sentence Embedding using InferSent

Parameters	Description
doc_words (list of strings)	Represents the input document as a list of words.
alpha (float, optional):	Denotes the learning rate utilized during inference, with a default value of 0.1.
steps (int, optional):	Specifies the number of steps (iterations) of inference to execute, defaulting to 5.
infer_subsample (float, optional):	Determines the subsampling threshold for common words during inference. If none is set, no subsampling occurs, with a default value of 0.1.
start_alpha (float, optional):	Refers to the initial learning rate. If not explicitly defined, it is automatically set to alpha, with a default of none.

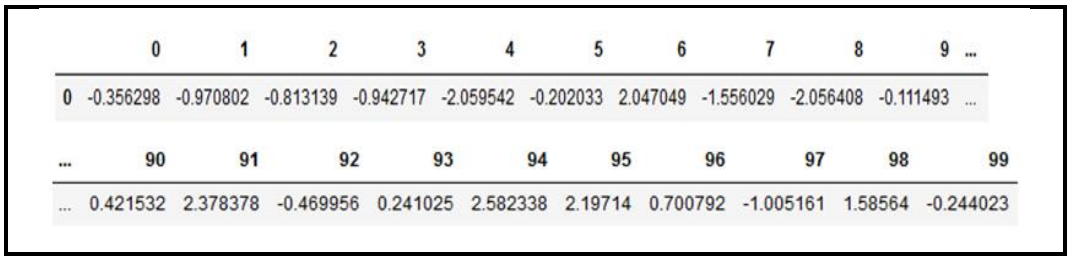


```
array([-0.356298  , -0.9708023 , -0.8131391 , -0.94271684, -2.0595422 ,
       -0.20203313,  2.047049  , -1.5560294 , -2.0564077 , -0.11149304,
       -1.1510105 , -1.3021445 , -0.01798234,  0.8491376 ,  1.3420774 ,
       -0.45485964,  2.5911045 ,  0.9078252 , -0.12256911,  0.04492952,
       -0.64614147, -1.0363629 ,  0.87329054, -0.8073886 ,  0.4469836 ,
       -2.2796252 ,  0.66568357, -0.36830965, -2.6160486 ,  0.58305174,
        2.5458844 ,  1.8198209 , -0.21816179,  1.8387325 , -1.2759304 ,
       -0.48836553,  0.66925275, -0.5990626 , -0.04637432,  0.09905079,
       -0.97831964, -0.29723758,  1.4903473 ,  1.8631814 , -0.7136153 ,
       -0.8199292 , -0.66341925,  0.5903203 ,  1.0379261 ,  0.2756216 ,
        1.1758718 ,  0.2343356 ,  0.9698275 , -1.2291751 , -0.5848837 ,
       -0.2261102 ,  1.7494977 ,  1.0212746 ,  0.0245354 ,  1.0990841 ,
        1.7408248 , -0.10135457, -1.7149194 , -0.547552  , -0.8317477 ,
        0.2622178 ,  0.23497152,  0.35079843, -0.837125  ,  0.29094574,
       -0.00624786, -2.7520332 ,  1.746678  , -0.7303855 ,  0.5894148 ,
        0.19358955, -0.33190817, -0.83852166,  0.31196588, -0.41932595,
        0.8356228 , -0.57885456, -0.24028236,  1.5787357 , -0.03365032,
       -0.94193125,  0.41075596, -0.35013926,  0.07900082, -1.0485425 ,
        0.4215319 ,  2.378378  , -0.4699563 ,  0.24102546,  2.5823376 ,
        2.19714   ,  0.7007924 , -1.0051608 ,  1.5856403 , -0.24402286],
      dtype=float32)
```

Figure 4.8 Example of Sentence Embedding Feature Set

4.6 Feature Combination and Concatenation

The sentence embedding features vector is merged or concatenated with the PSS features vector to create the proposed feature vector. In ISOT dataset, this combined feature vector has a size of (40199, 101) for training data and (467, 101) for testing data. The length of the first sentence in Figure 4.9 is one hundred dimensions. These features encapsulate the semantic and contextual information extracted from the text data. The Figure 4.10 illustrates the concatenated feature set, where the last value corresponds to the PSS score. This final feature encapsulates the probabilistic sentiment associated with each news article, providing valuable insight into its perceived authenticity. By integrating both sentence embedding and PSS features into the feature vector, our system captures a comprehensive representation of the textual data, enabling more robust and accurate classification of fake news. This combined feature set enriches the model's ability to discern subtle nuances and patterns in the data, ultimately enhancing its performance in distinguishing between real and fake news articles.



	0	1	2	3	4	5	6	7	8	9	...
0	-0.356298	-0.970802	-0.813139	-0.942717	-2.059542	-0.202033	2.047049	-1.556029	-2.056408	-0.111493	...

...	90	91	92	93	94	95	96	97	98	99
...	0.421532	2.378378	-0.469956	0.241025	2.582338	2.19714	0.700792	-1.005161	1.58564	-0.244023

Figure 4.9 Sentence Embedding Features for vector size “100”

	0	1	2	3	4	5	6	7	8	9	...
0	-0.356298	-0.970802	-0.813139	-0.942717	-2.059542	-0.202033	2.047049	-1.556029	-2.056408	-0.111493	...
...	91	92	93	94	95	96	97	98	99	100	
...	2.378378	-0.469956	0.241025	2.582338	2.19714	0.700792	-1.005161	1.58564	-0.244023	0.947345	

Figure 4.10 Concatenated Features (Sentence Embedding Features and PSS Feature)

4.7 Summary

This chapter embarks on a comprehensive exploration of our proposed model, delving deeply into its intricate details. The primary focus lies in elucidating the meticulous process through which we derive the PSS (Probabilistic Sentiment Score) from the carefully curated set of features. This extraction is accomplished by employing information gain as a pivotal criterion, ensuring the selection of the most informative attributes.Moreover, the methodology incorporates the utilization of InferSent, a sophisticated tool renowned for its proficiency in sentence embedding. Within this framework, each sentence is meticulously transformed into a rich 100-dimensional space, encapsulating its nuanced semantic essence. Here, the arsenal comprises SVM (Support Vector Machine) and NB (Naive Bayes) classifiers, esteemed for their prowess in handling diverse data sets with precision and efficacy.The culmination of these endeavors lies in the meticulous evaluation of the model's performance. Through rigorous experimentation and analysis , its efficacy and robustness are derived insights into. These evaluation results serve as the cornerstone for the comprehensive discussion awaiting in the subsequent chapter, wherein we delve deeper into the implications, limitations, and avenues for future research.

CHAPTER 5
EXPERIMENTAL RESULTS AND EVALUATION

This section presents the evaluation outcomes of the proposed framework in contrast to various frameworks designed for fake news classification. The experiment delves into the amalgamation of different features using two classifiers: Naive Bayes and Support Vector Machines (SVM) on the ISOT dataset and other fake news dataset from Kaggle.The experimental results of various models are meticulously analyzed, demonstrating the influence of PSS scores on sentence embedding features and TFIDF features across different classifiers. Furthermore, we conduct a comprehensive analysis of the performance metrics such as accuracy, precision, recall, and F1-score for each model. This evaluation allows us to compare the effectiveness of our framework with others in accurately classifying fake news. Through the experiment, it is used to provide valuable insights into the optimal feature combinations and classifier selections for improving the overall accuracy and reliability of fake news detection systems.

5.1 Model Evaluation

To evaluate the effectiveness of the proposed technique, various performance metrics such as accuracy, precision, recall, and F1-score can be computed using confusion matrices. Confusion matrices are especially useful in binary classification scenarios involving positive and negative classes, where True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) values are utilized to construct these matrices for each classifier.True Positive (TP) signifies the number of positive observations correctly classified by the model. False Negative (FN) represents positive instances incorrectly classified as negative. False Positive (FP) denotes negative instances mistakenly classified as positive. True Negative (TN) indicates the number of negative instances correctly classified. The accuracy of the model, calculated using Equation (5.1), quantifies the overall effectiveness of the model by measuring the percentage of correct predictions made across all instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$ Equation (5.1)

Precision, computed using Equation (5.2), evaluates the model's ability to correctly identify positive instances among all instances classified as positive.

$$Precision = \frac{TP}{TP+FP}$$ Equation (5.2)

Recall, determined by Equation (5.3), assesses the proportion of true positive predictions out of all actual positive instances in the dataset.

$$Recall = \frac{TP}{TP+FN}$$ Equation (5.3)

The F1-score, calculated using Equation (5.4), is a machine learning evaluation metric that combines both precision and recall, providing an overall measure of how well the model predicts across the entire dataset.

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall}$$ Equation (5.4)

These evaluation metrics derived from confusion matrices provide a comprehensive understanding of a classifier's performance, allowing for informed decisions regarding the efficacy and suitability of the proposed technique in handling binary classification tasks.

5.2 Proposed System Implementation using ISOT Dataset

This section describes the classification results of sentence embedding without PSS and with PSS based on ISOT Dataset.

5.2.1Classification Results of Sentence Embedding (without PSS) using SVM and NB Classifiers

The comprehensive analysis detailing the classification report of sentence embedding (without PSS) using SVM and NB in the tables below. This report encapsulates a thorough evaluation of the model's performance, including key metrics such as precision, recall, F1 score, and accuracy, providing valuable insights into its efficacy in classifying data instances within the given context.The table showcases a detailed breakdown of the model's performance across multiple classes or categories, highlighting its ability to

correctly identify and classify data points belonging to each category. Metrics such as precision illuminate the model's capability to accurately identify positive instances within a specific class, while recall measures its effectiveness in capturing all positive instances within that class. The F1 score, a harmonic mean of precision and recall, offers a balanced assessment of the model's overall performance. Additionally, the accuracy metric provides a comprehensive overview of the model's overall correctness in classifying data instances across all classes. This metric is particularly crucial as it reflects the model's ability to make accurate predictions and minimize misclassifications. Table 5.1 and Table 5.2 displays the test results of accuracy of classification result without PSS using SVM and NB .

Table 5.1 Classification Report for Sentence Embedding (without PSS) using SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.98	0.98	0.98	0.98
1	0.98	0.98	0.97	0.98

Table 5.2 Classification Report for Sentence Embedding (without PSS) using NB

Class	Accuracy	Precision	Recall	F1-Score
0	0.95	0.94	0.97	0.95
1	0.95	0.96	0.94	0.95

5.2.2 Classification Results of Sentence Embedding (with PSS) using SVM and NB Classifiers

The proposed system described the system sccuracy increase due to combined features (sentence embedding+PSS) in the previous chapter. The following Tables depict the classification report using these combined features.

Table 5.3 Classification Report for Sentence Embedding (with PSS) using SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.99	0.99	0.99	0.99
1	0.99	0.99	0.99	0.99

Table 5.4 Classification Report for Sentence Embedding (with PSS) using NB

Class	Accuracy	Precision	Recall	F1-Score
0	0.99	0.99	0.98	0.99
1	0.99	0.98	0.99	0.99

As depicted in Table 5.1, 5.2, 5.3, and 5.4, the proposed methods, SVM (with PSS) and NB (with PSS), exhibit superior performance in terms of accuracy, precision, and recall compared to NB (without PSS) and SVM (without PSS). Notably, all performance evaluations of the two classifiers with PSS surpass those without PSS values and the comparison results are depicted in Figure 5.1 and Figure 5.2. Additionally, the results indicate a consistent trend where SVM outperforms NB across all evaluation metrics. This discrepancy can be attributed to NB's limitations in handling sparse language data, primarily due to its exclusive reliance on co-occurrences with class labels, which may deviate from linguistic and semantic nuances.

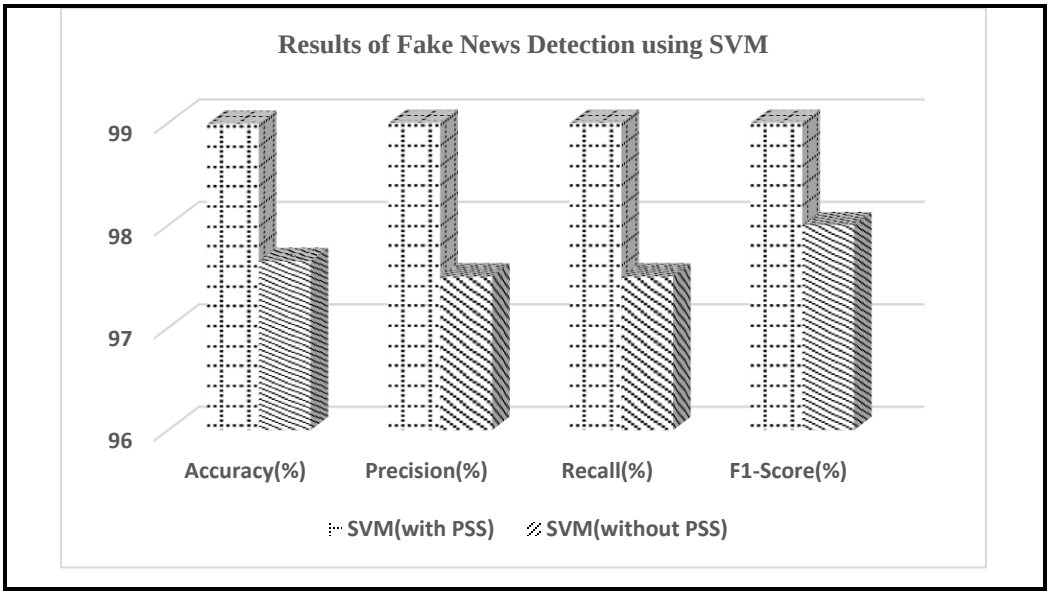


Figure 5.1 Results of Fake News Detection by SVM

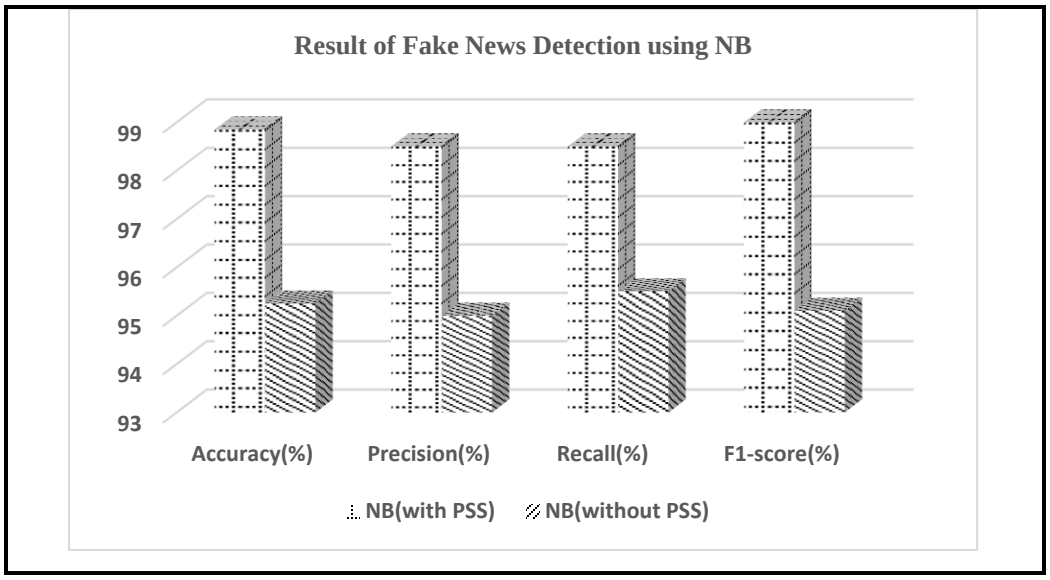


Figure 5.2 Results of Fake News Detection by NB

Furthermore, the proposed system implements sub models using other fake news dataset from Kaggle to highlight the contribution point in the next section.

5.3 Proposed System Implementation using other Fake News Dataset from Kaggle

This section implements the fake news detection system based on various features: TF-IDF, TF-IDF+PSS, Sentence embedding, Sentence embedding+PSS(without Infogain) and Sentence embedding+PSS(with Infogain) using SVM and NB classifiers and compare the classification results.

5.3.1 Classification Result of TF-IDF using SVM and NB Classifiers

The comprehensive analysis detailing the classification report for 3000,4000 and 5000 TFIDF features combined with SVM is visually presented in the table below. Table 5.5 displays the test results of accuracy at 84% for 3000 features, aiming to improve clarity and aid in understanding.

Table 5.5 Classification Report for TFIDF(3000) with SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.84	0.82	0.85	0.83
1	0.84	0.86	0.83	0.84

The following table shows the classification report for 4000 TFIDF features and the 82% accuracy attained using SVM.

Table 5.6 Classification Report for TFIDF(4000) with SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.82	0.80	0.90	0.85
1	0.82	0.89	0.60	0.72

The following table shows the classification report for 5000 TFIDF features and the 81% accuracy attained using SVM.

Table 5.7 Classification Report for TFIDF(5000) with SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.81	0.79	0.90	0.84
1	0.81	0.89	0.57	0.70

According to the comprehensive analysis presented in the classification report, it is evident that the SVM model exhibited its peak accuracy when integrated with 3000 TFIDF features while being tested on the dataset. This signifies a remarkable level of precision in classifying data instances, as clearly depicted in the accompanying figure. Nevertheless, this also hints at the potential presence of noise within the TFIDF features, indicating a necessity for their refinement through dimension reduction techniques to ensure optimal model performance.

The discerned high accuracy achieved with 3000 TFIDF features underscores their effectiveness in capturing relevant information and contributing significantly to the classification process. However, the identification of noise within these features necessitates a strategic approach to filter out irrelevant or redundant information. Dimension reduction methods such as information gain or feature selection algorithms can aid in streamlining the TFIDF feature space, thereby enhancing model efficiency and mitigating the impact of noise on classification outcomes.

Table 5.8 Classification Report for TFIDF(3000) with NB

Class	Accuracy	Precision	Recall	F1-Score
0	0.79	0.75	0.82	0.79
1	0.79	0.83	0.75	0.79

In contrast, while the SVM model showcased exceptional performance with 3000 TFIDF features, the presence of noise within these features emphasizes the importance of implementing dimension reduction strategies to optimize model accuracy and robustness in real-world applications.

As previously highlighted, it has been established that SVM demonstrates its highest performance when paired with TF-IDF 3000. Leveraging this insight, it is decided to apply TF-IDF 3000 in conjunction with Naïve Bayes to the test dataset. The resulting outcome of this model is depicted in the Figure 5.3 below, providing a visual representation of the performance metrics and comparative analysis between the various configurations tested. This approach enables us to assess the effectiveness and suitability of TF-IDF 3000 in combination with Naïve Bayes, shedding light on its performance in real-world testing scenarios and contributing valuable insights to the ongoing exploration of optimal classification methodologies.

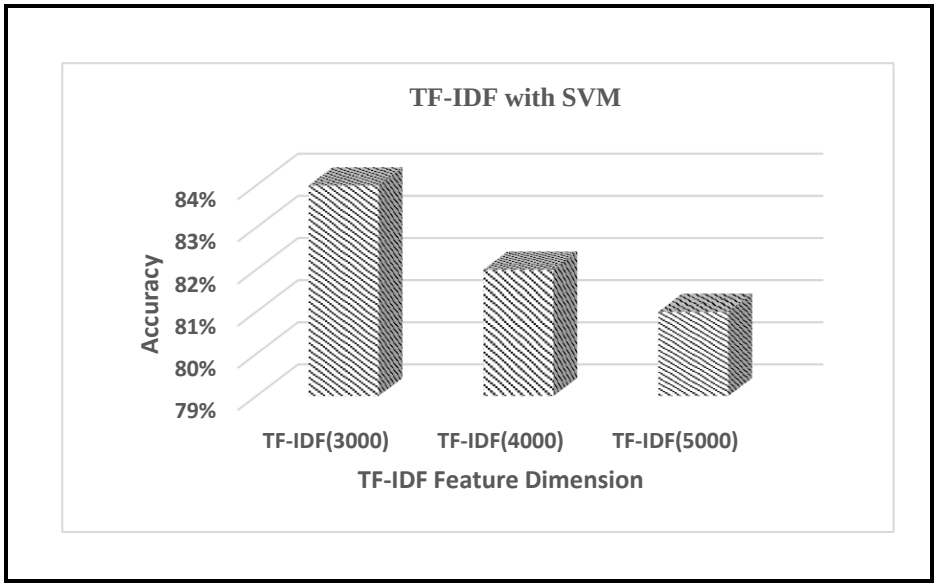


Figure 5.3 Classification results with each TF-IDF Features Dimension

The experimental results clearly highlight the superior performance of utilizing TFIDF 3000 in conjunction with SVM, showcasing an impressive accuracy rate of 84%. This outcome stands out as it surpasses the accuracies achieved with TFIDF 4000 and 5000 configurations. Notably, TFIDF 3000 combined with SVM outperforms TFIDF used alongside Naïve Bayes. This finding underscores the efficacy and potential advantages of leveraging SVM in tandem with TFIDF 3000 for classification tasks, showcasing its ability to yield more precise and reliable results compared to alternative approaches.

5.3.2 Classification Result of TF-IDF (with PSS) using SVM and NB Classifiers

This section delves into the impressive results achieved through SVM classification utilizing TFIDF in conjunction with PSS, boasting an outstanding accuracy of 87%, as vividly illustrated in Table 5.9. Building upon our earlier discussion of SVM's performance when utilizing TFIDF alone, which yielded a commendable accuracy of 84%, the incorporation of PSS into the SVM-based TFIDF classification process has resulted in a significant 3% increase in accuracy. This notable improvement serves as a testament to the efficacy and utility of PSS in augmenting the predictive capabilities of the SVM model, underscoring its invaluable contribution to the classification process.

Table 5.9 Classification result of SVM using TFIDF with PSS

Class	Accuracy	Precision	Recall	F1-Score
0	0.87	0.85	0.87	0.86
1	0.87	0.88	0.87	0.87

This section examines the outcomes of employing NB for classification with TFIDF in conjunction with PSS, resulting in a notable accuracy rate of 81%, as illustrated in the Table 5.10. Comparatively, as outlined in the prior section, NB utilization of TFIDF alone yielded an accuracy of 79%. Consequently, integrating PSS into the NB-based TFIDF classification process led to a substantial 2% increase in accuracy. This enhancement underscores the efficacy of incorporating PSS to enhance the predictive capabilities of the NB model, emphasizing its effectiveness within the classification framework. Moving

forward, we will delve deeper into the specific mechanisms through which PSS contributes to these improvements and explore potential avenues for further optimization.

Table 5.10 Classification Result of NB using TFIDF with PSS

Class	Accuracy	Precision	Recall	F1-Score
0	0.81	0.78	0.85	0.81
1	0.81	0.85	0.78	0.81

As the progress, the focus will shift towards a deeper examination of the specific mechanisms through which PSS contributes to these substantial accuracy improvements. Additionally, potential avenues will be explored for further optimization to harness the full potential of PSS in refining and enhancing classification models for diverse applications.

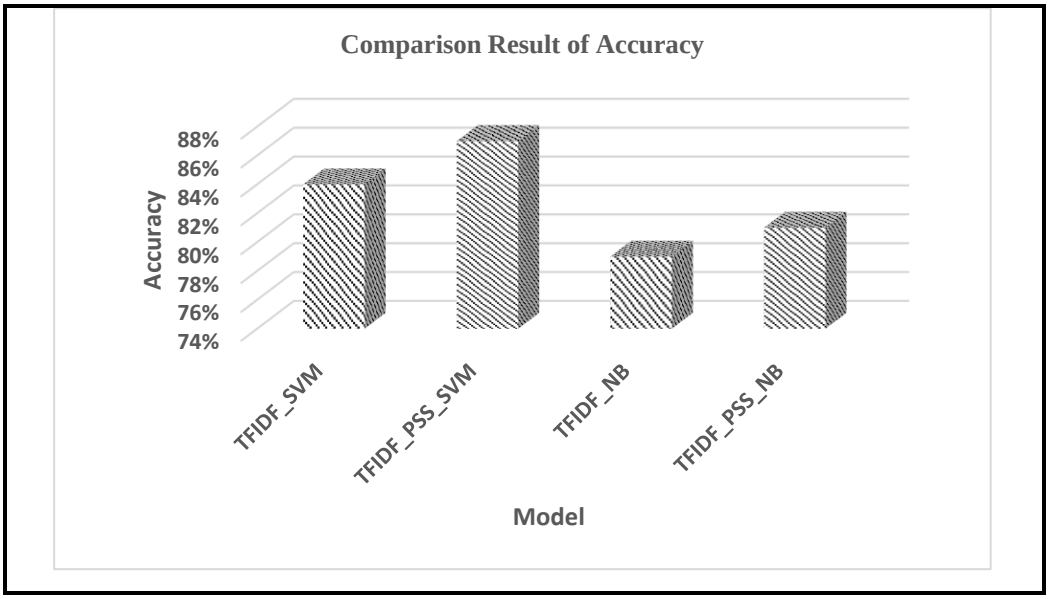


Figure 5. 4 Compare the Results of TF-IDF and Combined TF-IDF and PSS with SVM and NB

As visually depicted in the Figure 5.4, the incorporation of PSS has demonstrated a significant enhancement in the performance of TFIDF features when utilized in conjunction with both SVM (Support Vector Machine) and NB (Naïve Bayes) classification models. Specifically, PSS has been instrumental in improving the accuracy of TFIDF features by 3% when paired with SVM and by 2% when integrated with NB.

This substantial improvement underscores the effectiveness of PSS in refining the predictive capabilities of these classification models. By leveraging PSS, the models can better discern patterns and relationships within the feature space, leading to more accurate and reliable classification outcomes. The 3% gain in accuracy observed with SVM and the 2% enhancement with NB highlight the versatility of PSS in positively impacting various classification algorithms, irrespective of their underlying mechanisms.

Furthermore, this enhancement in accuracy showcases the potential of incorporating advanced sampling techniques like PSS to augment the performance of machine learning models, particularly in scenarios where feature space complexities can hinder traditional classification approaches. The results presented in the figure validate the importance of considering innovative strategies like PSS to optimize model performance and enhance the overall efficacy of classification systems in real-world applications.

5.3.3 Classification Result of Sentence Embedding (without PSS) using SVM and NB Classifiers

SVM classification utilizing sentence embedding, but without the inclusion of a probabilistic sentiment score (PSS), achieves an impressive accuracy rate of 90% in Table 5.11. Similarly, when employing Naïve Bayes (NB) for classification under the same conditions, an accuracy level of 85% is attained in Table 5.12. These results highlight the robustness and effectiveness of both SVM and NB in accurately categorizing data instances based on sentence embeddings, showcasing their capability to handle complex linguistic features and nuances within textual data.

Table 5.11 Classification Result of SVM using Sentence Embedding without a Probabilistic Sentiment Score (PSS)

Class	Accuracy	Precision	Recall	F1-Score
0	0.90	0.87	0.93	0.90
1	0.90	0.93	0.87	0.90

Expanding on these findings, the high accuracy achieved by SVM and NB underscores their suitability for tasks requiring sophisticated text analysis and sentiment

classification. The utilization of sentence embeddings, which capture semantic and contextual information from text, further enhances the models' ability to discern subtle variations in sentiment and meaning, leading to more precise classification outcomes.

Moreover, these results emphasize the importance of considering different classification algorithms and feature representations when dealing with natural language processing tasks. By exploring various techniques such as sentence embedding and model-specific enhancements like PSS, researchers and practitioners can unlock new avenues for improving the accuracy and robustness of sentiment analysis and text classification systems. The successful classification outcomes achieved by SVM and NB using sentence embedding without PSS underscore their efficacy in handling complex textual data, paving the way for advancements in fake news detection and related NLP applications.

Table 5.12 Classification Result of NB using Sentence Embedding without a Probabilistic Sentiment Score (PSS)

Class	Accuracy	Precision	Recall	F1-Score
0	0.85	0.82	0.86	0.84
1	0.85	0.87	0.83	0.85

The Figure 5.5 illustrates a comparative analysis of the experimental outcomes achieved using TFIDF features and sentence embedding features in conjunction with SVM and NB classifiers. It is noteworthy to mention that in this model setup, the probabilistic sentiment score (PSS) was not utilized alongside the sentence embedding (SE) features.

Upon analyzing the results, it is observed that both SVM and NB classifiers exhibit an improvement of 6% in accuracy when using sentence embedding features compared to TFIDF features alone. This finding highlights the potential of sentence embedding as a feature representation method for enhancing the performance of classification models.

The utilization of sentence embedding features offers several advantages, such as capturing semantic information and contextual nuances, which are often missed by traditional TFIDF-based approaches. This richer feature representation enables the classifiers to make more informed decisions, resulting in improved accuracy rates.

Furthermore, the absence of PSS in conjunction with sentence embedding suggests that the inherent features extracted from the text alone are sufficiently informative for achieving notable accuracy gains. However, it is worth exploring how incorporating PSS alongside sentence embedding features could further enhance the model's predictive capabilities and whether it leads to additional improvements in accuracy.

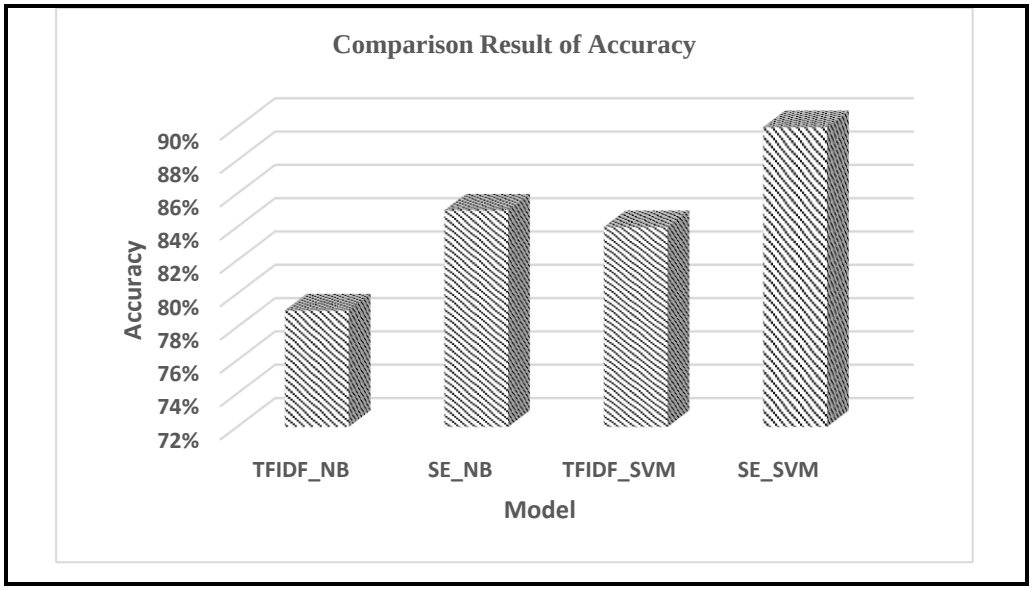


Figure 5.5 The Comparison Results of TFIDF standalone with Sentence Embedding using SVM and NB

Overall, the comparison presented in the Figure 5.5 underscores the effectiveness of sentence embedding features in conjunction with SVM and NB classifiers, showcasing their potential to elevate the performance of fake news classification tasks.

5.3.4 Classification Result of Sentence Embedding (with PSS) using SVM and NB Classifiers

SVM classification, which integrates Support Vector Machines with advanced sentence embedding techniques and incorporates a Probabilistic Sentiment Score (PSS), showcases remarkable performance, achieving an impressive accuracy rate of 90% as depicted in Table 5.13. Similarly, when applying Naïve Bayes (NB) for classification under identical conditions, it also attains a high accuracy level of 87%, as illustrated in Table 5.14. An interesting observation from our analysis is that the inclusion of PSS has a positive

impact on the sentence embedding features. More precisely, PSS can increase the accuracy of NB by 2%, but it does not have any effect on SVM because the accuracies remain the same whether or not PSS classification is used with SVM.

Table 5.13 Classification of SVM using Sentence Embedding with a Probabilistic Sentiment Score

Class	Accuracy	Precision	Recall	F1-Score
0	0.90	0.87	0.93	0.90
1	0.90	0.93	0.88	0.90

These results underscore the robustness and effectiveness of NB methodologies in handling sentiment analysis tasks, especially when leveraging sophisticated techniques like sentence embedding and integrating PSS into the classification process. This emphasizes the reliability and efficiency of these approaches in accurately categorizing fake news in textual data, making them valuable tools for extracting meaningful insights from large volumes of textual content.

Table 5.14 Classification of NB using Sentence Embedding with a Probabilistic Sentiment Score

Class	Accuracy	Precision	Recall	F1-Score
0	0.87	0.85	0.88	0.87
1	0.87	0.89	0.86	0.88

The overall performance depicted in the Figure 5.6 serves to emphasize and highlight the efficacy of utilizing sentence embedding features in combination with Probabilistic Sentiment Score (PSS), both with and without PSS, in tandem with SVM and NB classifiers. This comparison underscores their collective potential to significantly enhance the performance and accuracy of fake news classification tasks. By integrating sentence embedding features, the classifiers can better capture the nuanced semantics and contextual information present in the text, enabling more precise and reliable classification of fake news articles. The inclusion of PSS further augments this capability by providing additional sentiment-based insights, which can serve as valuable indicators in distinguishing between genuine and misleading content.

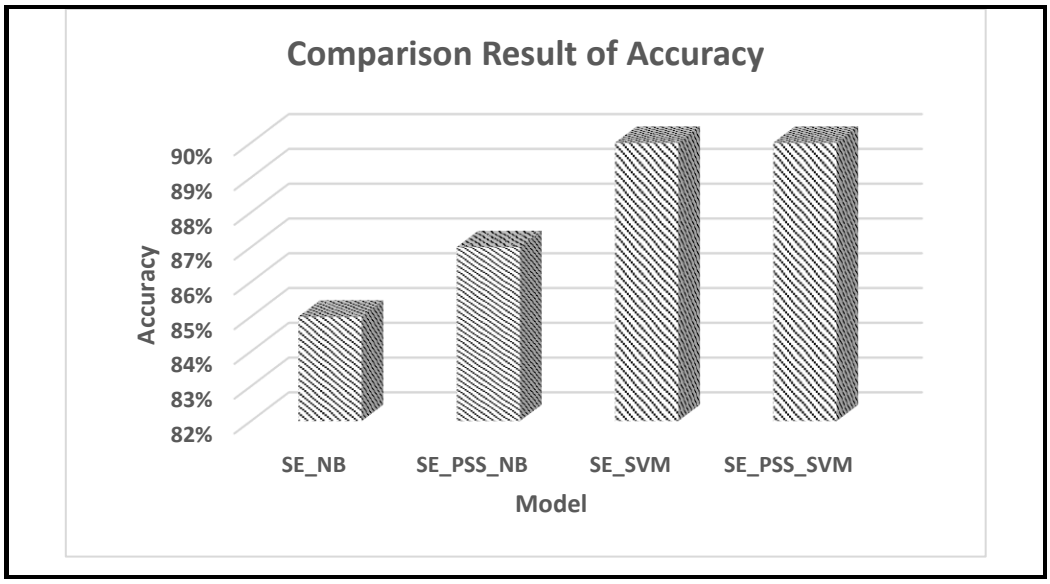


Figure 5.6 Comparison Results of Sentence Embedding, Sentence Embedding with PSS using SVM and NB

The comparison presented in the figure 5.6 serves as a testament to the effectiveness of these advanced techniques in elevating the sophistication and accuracy of fake news detection systems. It showcases the significant strides made in leveraging cutting-edge technologies to combat the spread of misinformation and enhance the trustworthiness of information dissemination platforms.

5.3.5 The Impact of Information Gain from the PSS on Sentence Embedding

In the previous discussion, the complexities involved in computing the Probabilistic Sentiment Score (PSS) were explored by leveraging TFIDF features alongside logistic regression. The TFIDF feature dimension represents the breadth of the dataset's lexicon, which, in the specific case, encompasses a vast array of features. To assess the relevance of each feature to the target, we employed information gain, resulting in a selection of 3000 features with their respective infogain values depicted in the accompanying figure. The criterion for feature selection was an infogain score exceeding 0.1, leading us to identify and retain 100 features out of the initial pool. The Figure 5.7 illustrates the indices of these 100 features.

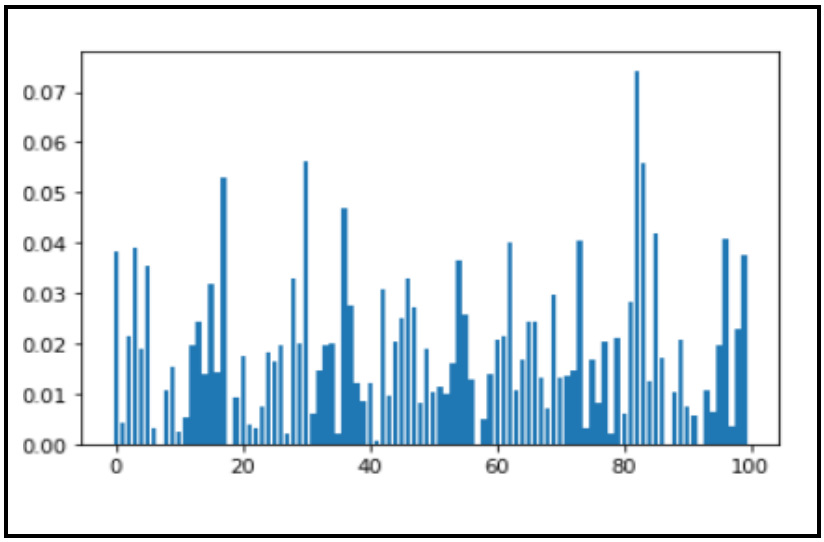


Figure 5.7 Information Gain Score of the best 100 features

When PSS scores were initially employed from TFIDF 3000 features for fake news classification using sentence embedding classify with the Naïve Bayes (NB) and SVM algorithms, the system achieved a commendable accuracy rate of 87% and 90%. However, given the vastness of this feature set, it is likely that certain features within it may not significantly contribute to the classification outcomes.

To tackle this potential issue and streamline the classification process, the concept of information gain is implemented to reduce the feature dimensionality, specifically for computing the PSS score.

Table 5.15 Classification Result of NB

Class	Accuracy	Precision	Recall	F1-Score
0	0.87	0.86	0.87	0.86
1	0.87	0.88	0.87	0.87

By selecting the top 100 features with the highest information gain, we noticed no difference in accuracy, with the NB and SVM-based classifications achieving an accuracy rate of 87% and 90% as shown in Table 5.15 and 5.16, respectively, compared to the PSS score without information gain. This finding underscores the efficacy of employing feature selection techniques like information gain to bolster model performance by focusing on the most informative features while discarding less relevant ones.

Table 5.16 Classification Result of SVM

Class	Accuracy	Precision	Recall	F1-Score
0	0.90	0.87	0.93	0.90
1	0.90	0.93	0.88	0.90

By leveraging selected features with high information gain, it can not only maintain accuracy but also reduce the time complexity of PSS model training. By utilizing specific characteristics that have a significant impact on the amount of useful information, it may not only preserve the precision but also decrease the computational complexity of training the PSS model. Despite our research not specifically exploring the time-consuming aspect, the focus remained on showcasing the precision exhibited by our models. This led to the development of a hypothesis suggesting that the computation of the PSS score could be effectively accomplished using a subset of features. In this context, the recommendation leans towards selecting the top 100 features as optimal for the task. By doing so, it is necessary to avoid incorporating the entire to incorporate the entire feature set, as our findings suggest that the inclusion of all features isn't necessary to maintain satisfactory PSS performance. This insight not only highlights the potential for efficiency gains in computational resources but also underscores the robustness of our model, demonstrating its ability to deliver accurate results while operating with reduced complexity.

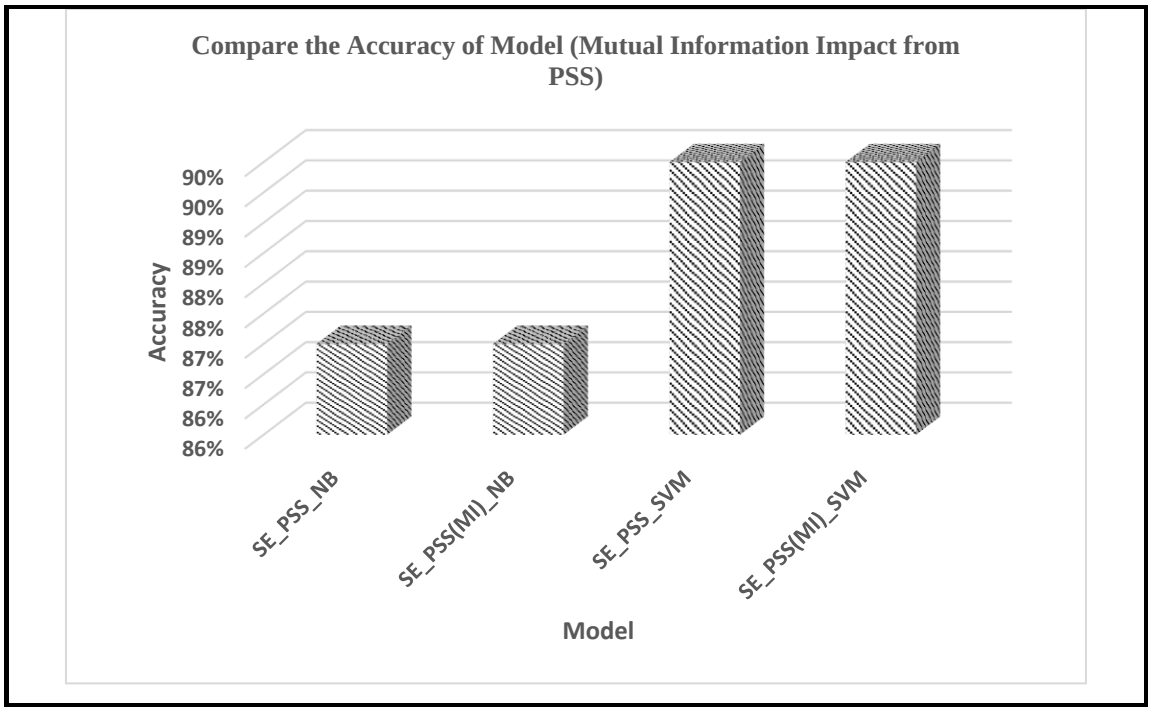


Figure 5. 8 Mutual Information (MI) Impact from PSS Score

Figure 5.8 presented the detail of the accuracy metrics of various models, highlighting the standout performers in the analysis. The most notable models are those that integrate sentence embeddings with Probabilistic Sentiment Scores (PSS) and are classified by Support Vector Machines (SVM) and Naïve Bayes (NB), achieving an impressive accuracy rate of 90% each.Furthermore, the exploration delves into the impact of PSS scores on model accuracy, particularly when used in conjunction with different features. It is found that solely incorporating PSS scores without considering information gain does not significantly enhance model accuracy, especially when compared to the efficacy of utilizing sentence embedding features.

However, the integration of PSS scores with TFIDF (Term Frequency-Inverse Document Frequency) features and subsequent classification by SVM demonstrates a remarkable improvement in accuracy, achieving an impressive 87% accuracy rate. This finding underscores the synergistic effect of combining fake news analysis with traditional text features, leading to enhanced performance in classification tasks.

Overall, the analysis emphasizes the importance of feature selection and integration strategies in achieving optimal model accuracy. By leveraging advanced techniques such

as sentence embeddings, probabilistic sentiment scores, and appropriate classification algorithms like SVM and NB, it can significantly enhance the accuracy and effectiveness of machine learning models in tasks such as fake news detection.

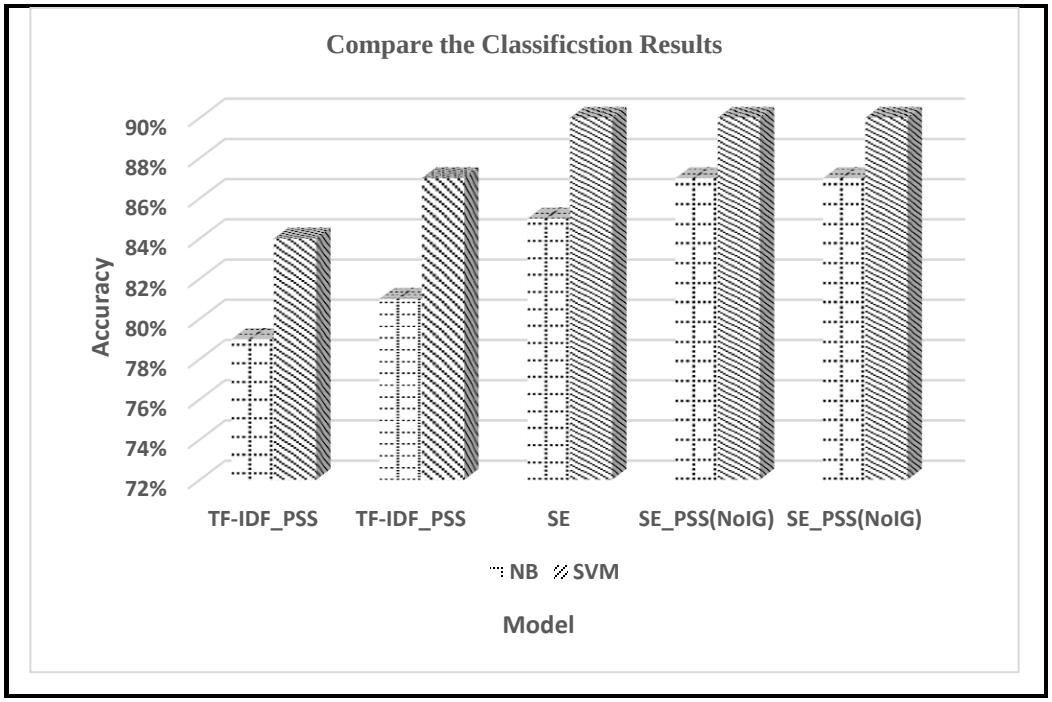


Figure 5.9 Comparison of Different Model's Accuracy

5.4 Comparison of Classification Results between ISOT and other Fake News Dataset

The impact of Probabilistic Sentiment Scores (PSS) is analysed on sentence embedding features across various datasets, as illustrated in the following Figure 5.10. The study involves evaluating how integrating PSS with sentence embeddings affects the performance of classification models. By comparing results across two datasets: ISOT and other fake news dataset. It aims to determine the consistency and effectiveness of this approach. The figure provides a visual representation of the performance metrics, highlighting the improvements in accuracy, precision, and recall achieved by incorporating PSS scores. The comprehensive analysis helps to underscore the potential benefits of combining sentiment analysis with sentence embeddings in enhancing model performance.

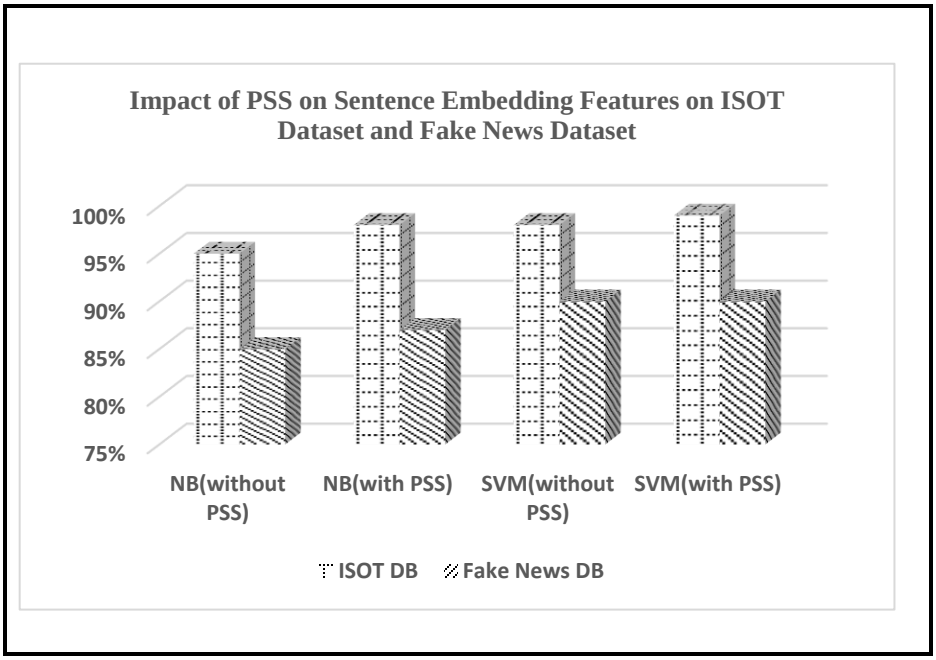


Figure 5.10 The impact of PSS on Sentence Embedding features on ISOT Dataset and Fake News Dataset

As illustrated in the above figure, the impact of Probabilistic Sentiment Score (PSS) on sentence embedding features when using NB on the ISOT dataset results in a 3% improvement in accuracy. For the other fake news dataset, the improvement achieved is 2%. When applying SVM on the ISOT dataset, there is a 1% improvement in accuracy with the inclusion of PSS compared to without PSS. However, on the other fake news dataset, the inclusion of PSS does not result in a significant improvement. This indicates that while PSS enhance the performance of NB and SVM on the ISOT dataset, its effect is less pronounced on the other fake news dataset.

5.5 Summary

This chapter delves into the experimental outcomes obtained from employing various models that combine different features for classification using Support Vector Machine (SVM) and Naïve Bayes (NB) using ISOT dataset and other small fake news dataset from Kaggle.

In ISOT dataset, the proposed methods: NB (with PSS) and SVM (with PSS) could stand tallest around (98.46%,98.99%), (98.5% ,99%) and (98.5% ,99%) in accuracy,

precision and recall rather than NB (without PSS) and SVM (without PSS). All performance evaluations of two classifiers with PSS are better than these two classifiers without PSS values. Moreover, based on the results, we can also conclude that SVM performs better than NB in all evaluation metrics. Because, NB cannot handle well sparsely of languages due to its solely examination of co-occurrences with class labels, which is a little far precision from the language linguistics and semantic concepts.

Moreover, the proposed system highlights the contribution using another fake news dataset with various features. Initially, the results are presented which are obtained from utilizing 3000, 4000, and 5000 TFIDF features with SVM and NB. Among these, the TFIDF 3000 feature set achieves the highest accuracy at 84%, surpassing the accuracies of 82% for 4000 TFIDF features and 81% for 5000 TFIDF features. From these results, it is a deduction that TFIDF features may not consistently perform well in high-dimensional spaces due to the presence of noise features, necessitating the use of feature dimensionality reduction techniques.

Next, the influence of Probabilistic Sentiment Score (PSS) is analysed on TFIDF feature classification with SVM and NB. It is demonstrated that integrating the PSS score can enhance SVM accuracy by 3%, resulting in 87% accuracy, and NB accuracy by 2%, reaching 81%. The research reveals that the PSS score, derived from TFIDF features using a logistic regression model, already encapsulates target information, which is then utilized as an input feature, thereby positively impacting the classification outcome.

Moving on, sentence embedding features are explored with SVM and NB, achieving 90% and 85% accuracy, respectively. These results indicate a 6% accuracy improvement over TFIDF features for both classifiers. Sentence embedding features outperform TFIDF features due to their utilization of word embedding (such as doc2vec) for semantic information and treating each sentence as a feature, thereby incorporating contextual information in addition to statistical considerations.

In addition, it is shown that combining the PSS score with phrase embedding features results in a 90% accuracy for SVM and a 90% accuracy for NB. This demonstrates a 2% improvement for SVM compared to other models, while NB maintains the same accuracy without the PSS feature. Additionally, the investigation of the impact of

information gain score derived from PSS are done on model accuracy. While information gain scores do not enhance accuracy, they can decrease the complexity of features and time.

In summary, the research highlights the positive impact of PSS score on TFIDF features and sentence embedding features when classified with SVM and NB. Moreover, we emphasize the importance of infogain score for PSS features in enhancing model accuracy. In the subsequent chapter, the discussion of the conclusions drawn from the research and outline future directions for exploration are described.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter describes into the conclusion of the research, which centers on the development of a novel fake news detection framework. The approach utilizes a combination of probabilistic sentiment scores (PSS) as a feature alongside sentence embedding features, aiming to overcome the limitations of existing fake news detection systems that primarily rely on word embedding features. Traditional fake news detection systems often utilize word embedding features, but these approaches may struggle with semantic nuances and lack coverage of contextual features. To address these challenges, our research introduces the use of sentence embedding features, which capture a broader context and provide a more comprehensive understanding of the text. InferSent, a tool developed by Facebook is leveraged, to generate high-quality sentence embeddings from the textual data.

Additionally, an idea is contributed by incorporating probabilistic sentiment scores as a feature. These scores are derived from a logistic regression classifier, which classifies the sentiment of the text. This sentiment classification result is then integrated with the sentence features, enhancing the overall detection capability of our framework.To evaluate the effectiveness of the proposed method, experiments are conducted using two classification models, including Support Vector Machines (SVM) and Naive Bayes (NB), on the ISOT dataset and other fake news dataset for fake news detection research. The results indicate that the approach, combining sentence embedding features with PSS and employing NB and SVM classifiers, achieves the highest accuracy rate of 99%, outperforming the model without PSS.

Furthermore, the research goes beyond accuracy metrics and delves into analyzing the impact of PSS scores on traditional TFIDF (Term Frequency-Inverse Document Frequency) features and sentence embeddings. The influence of PSS scores is explored on the classification results using information gain analysis, providing a comprehensive understanding of how sentiment analysis can enhance fake news detection capabilities.

Overall, the study showcases the effectiveness of incorporating probabilistic sentiment scores and sentence embeddings in fake news detection frameworks,

demonstrating significant improvements in accuracy and highlighting the importance of considering contextual and sentiment-based features in text classification tasks.

Classification models using Support Vector Machine (SVM) and Naive Bayes (NB) with various feature combinations. Initially, results from TFIDF feature sets of different dimensions are presented, emphasizing the need for dimensionality reduction techniques due to noise features in high-dimensional spaces. The influence of Probabilistic Sentiment Score (PSS) on TFIDF feature classification with SVM and NB is analyzed, showcasing notable accuracy improvements when integrating PSS scores. Furthermore, the exploration of sentence embedding features demonstrates their superiority over TFIDF features, leveraging semantic and contextual information for enhanced accuracy.

The integration of PSS scores with sentence embedding features yields significant accuracy improvements for both SVM and NB classifiers. Additionally, the impact of information gain score derived from PSS on accuracy is investigated, highlighting its beneficial role in model performance. In summary, the research emphasizes the positive impact of PSS scores on TFIDF and sentence embedding features when classified with SVM and NB, emphasizing the importance of infogain for PSS features. The abstract hints at further discussions in future chapters regarding conclusions drawn and potential directions for future research in this domain.

6.1 Advantages and Limitations

Although the research has achieved remarkable results using sentence embedding integrated with probabilistic sentiment scores and classified with Support Vector Machines (SVM) and Naive Bayes (NB), there are still some limitations that need to be addressed as following.

Subtle linguistic nuances: Using sentence embedding features for fake news detection presents several limitations. Firstly, there is a risk of losing detailed information and context as sentence embeddings focus on capturing the overall semantic meaning of a sentence. This approach may overlook subtle linguistic nuances, word-level inconsistencies, or specific textual patterns that could signal fake news.

Interpretability: One of the limitations is the interpretability of the model. While the combined approach may yield high accuracy, understanding why certain decisions are

made by the model can be challenging. This lack of interpretability can be a drawback, especially in critical applications where transparency is crucial.

Domain-Specific Adaptation: The effectiveness of the model may vary across different domains or topics. It is important to assess the model's performance in various contexts to ensure its generalizability and adaptability to different types of fake news.

Scalability: As the dataset size increases, the computational resources required to train and deploy the model also increase. Ensuring scalability and efficiency, especially in real-time applications or with large-scale datasets, is a challenge that needs to be addressed.

Robustness to Adversarial Attacks: Fake news generators may actively try to evade detection by crafting misleading content specifically designed to fool the model. Enhancing the model's robustness against such adversarial attacks is an ongoing research challenge.

6.2 Future Work

While the research has indeed achieved good accuracy in fake news detection, there are still several areas we need to address. Firstly, the applicability of the model must be expanded by testing it on a Myanmar news dataset to ensure its effectiveness across different linguistic and cultural contexts. Additionally, it is necessary to explore the capabilities of end-to-end models such as the transformer model to enhance the robustness and scalability of the approach. These steps are crucial for advancing the reliability and generalizability of the fake news detection system.

AUTHOR'S PUBLICATIONS

[p1] May Me Me Hlaing, Nang Saing Moon Kham “Fake News Detection Using Machine Learning Approach”, The 1st Annual University Journal on Research and Application (AUJRA),pp. 41-57,2019.

[p2] May Me Me Hlaing, Nang Saing Moon Kham “Defining News Authenticity on Social Media Using Machine Learning Approach”, The 18th International Conference on Computer Application (ICCA 2020),pp. 123-129,2020.

[p3] May Me Me Hlaing, Nang Saing Moon Kham “Classifying Social Media News Authenticity Using Machine Learning and Ensemble Methods”, The 2nd Myanmar Universities’ Research Conference (MURC 2020),pp. 16,2020

[p4] May Me Me Hlaing, Nang Saing Moon Kham “Comparative Study of Fake News Detection Using Machine Learning and Neural Network Approaches”, The 13th International Conference on Future Computer and Communication (ICFCC 2021)), pp. ,59-64,2021

[p5] May Me Me Hlaing, Win Lelt Lelt Phyu “Detecting Fake News Through Probabilistic Sentiment Model And Sentence Embedding Technique”, Indian Journal of Computer Science and Engineering”, Vol. 15, No 1,pp., 68-79,2024, DOI : 10.21817/indjcse/2024/v15i1/241501024

BIBLOGRAPHY

[1] Kim, Jooyeon, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. "Leveraging the crowd to detect and reduce the spread of fake news and misinformation." In Proceedings of the eleventh ACM international conference on web search and data mining, pp. 324-332. 2018.

[2] Rubin, Victoria L. "Deception detection and rumor debunking for social media." In The SAGE handbook of social media research methods, p. 342. London: Sage, 2017.

[3] Volkova, S., & Jang, J. Y. (2018). Misleading or Falsification: Inferring Deceptive Strategies and Types in Online News and Social Media. The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018, 575–583. https://doi.org/10.1145/3184558.3188728

[4] Zhang, Xichen, and Ali A. Ghorbani. "An overview of online fake news: Characterization, detection, and discussion." Information Processing & Management 57, no. 2 (2020): 102025.

[5] Nasir Ansari, J. A., Mohd Azhar, & Akhtar, M. J. (2022). The spread of Misinformation on social media: An insightful countermeasure to restrict. Studies in Economics and Business Relations, 3(1). https://doi.org/10.48185/sebr.v3i1.401

[6] Sepúlveda-Torres, Robiert, Alba Bonet-Jover, and Estela Saquete. "Detecting misleading headlines through the automatic recognition of contradiction in spanish." IEEE Access (2023).

[7] Nagi, Kuldeep. "New social media and impact of fake news on society." ICSSM Proceedings, July (2018): 77-96.

[8] Tandoc Jr, Edson C. "The facts of fake news: A research review." Sociology Compass 13, no. 9 (2019): e12724.

[9] Zubiaga, Arkaitz, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. "Detection and resolution of rumours in social media: A survey." Acm Computing Surveys (Csur) 51, no. 2 (2018): 1-36.

[10] Li, Quanzhi, Qiong Zhang, Luo Si, and Yingchi Liu. "Rumor detection on social media: Datasets, methods and opportunities." arXiv preprint arXiv:1911.07199 (2019).

[11] Pathak, Ajeet Ram, Aditee Mahajan, Keshav Singh, Aishwarya Patil, and Anusha Nair. "Analysis of techniques for rumor detection in social media." Procedia Computer Science 167 (2020): 2286-2296.

[12] Kumar, Akshi, and Saurabh Raj Sangwan. "Rumor detection using machine learning techniques on social media." In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2, pp. 213-221. Springer Singapore, 2019.

[13] Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. In Information (Switzerland) (Vol. 10, Issue 4). MDPI AG. https://doi.org/10.3390/info10040150

[14] Gasparetto, Andrea, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. "A survey on text classification algorithms: From text to predictions." Information 13, no. 2 (2022): 83.

[15] Khan, Aurangzeb, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. "A review of machine learning algorithms for text-documents classification." Journal of advances in information technology 1, no. 1 (2010): 4-20.

[16] Qader, Wisam A., Musa M. Ameen, and Bilal I. Ahmed. "An overview of bag of words; importance, implementation, applications, and challenges." In 2019 international engineering conference (IEC), pp. 200-204. IEEE, 2019.

[17] Henno, J. (n.d.). Information and information security Information and Information Security . http://ceur-ws.org

[18] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. In IEEE Access (Vol. 7, pp. 53040–53065). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ ACCESS. 2019.2912200

[19] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9, no. 1 (2020): 381-386.

[20] Steinwart, Ingo, and Andreas Christmann. Support vector machines. Springer Science & Business Media, 2008.

[21] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detection of online fake news using n-gram analysis and machine learning techniques." In Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1, pp. 127-138. Springer International Publishing, 2017.

[22] Goksu, Murat, and Nadire Cavus. "Fake news detection on social networks with artificial intelligence tools: systematic literature review." In International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions, pp. 47-53. Cham: Springer International Publishing, 2019.

[23] Paul, Shovon, Jubair Islam Joy, Shaila Sarker, Sharif Ahmed, and Amit Kumar Das. "Fake news detection in social media using blockchain." In 2019 7th international Conference on smart computing & communications (ICSCC), pp. 1-5. IEEE, 2019.

[24] Ivanov, Stanislav Hristov. "Automated decision-making." foresight 25, no. 1 (2023): 4-19.

[25] Ghosh, Moumita, and A. Thirugnanam. "Introduction to Artificial Intelligence." Artificial Intelligence for Information Management: A Healthcare Perspective (2021): 23-44.

[26] Muslim, Eman M. "An Introduction to Computational Linguistics Advantages&Disadvantages." journal of the college of basic education 51 (2007): 29-40.

[27] Nadkarni, Prakash M., Lucila Ohno-Machado, and Wendy W. Chapman. "Natural language processing: an introduction." Journal of the American Medical Informatics Association 18, no. 5 (2011): 544-551.

[28] Heines, Roger, Christian Dick, Christian Pohle, and Reinhard Jung. "The Tokenization of Everything: Towards a Framework for Understanding the Potentials of Tokenized Assets." In PACIS, p. 40. 2021.

[29] Myint, Cynthia. "A Part of Speech Tagger for Myanmar Text." PhD diss., MERAL Portal, 2011.

[30] Pickering, Martin J., and Roger PG Van Gompel. "Syntactic parsing." In Handbook of psycholinguistics, pp. 455-503. Academic Press, 2006.

[31] Goddard, Cliff, and Andrea C. Schalley. "Semantic Analysis." (2010): 93-120.

[32] Dobson, Annette J. Introduction to statistical modelling. Springer, 2013.

[33] Sanderson, Mark, and Justin Zobel. "Information retrieval system evaluation: effort, sensitivity, and reliability." In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 162-169. 2005.

[34] Nagarhalli, Tatwadarshi P., Vinod Vaze, and N. K. Rana. "A review of current trends in the development of chatbot systems." In 2020 6th International conference on advanced computing and communication systems (ICACCS), pp. 706-710. IEEE, 2020.

[35] Duizith, José Luiz Andrade, Lizandro Kirst da Silva, Daniel Ribeiro Brahm, Gustavo Tagliassuchi, and Stanley Loh. "A virtual assistant for websites." Revista Eletronica de Sistemas de Informaçao 3, no. 1 (2004).

[36] Kulagina, Olga S., and Igor A. Mel'čuk. "Automatic translation: some theoretical aspects and the design of a translation system." (2003).

[37] Roy, Khushi, Subhra Debdas, Sayantan Kundu, Shalini Chouhan, Shivangi Mohanty, and Biswarup Biswas. "Application of natural language processing in healthcare." Computational Intelligence and Healthcare Informatics (2021): 393-407.

[38] Roos, Inger, and Bo Edvardsson. "Customer-support service in the relationship perspective." Managing Service Quality: An International Journal 18, no. 1 (2008): 87-107.

[39] Basheer, Imad A., and Maha Hajmeer. "Artificial neural networks: fundamentals, computing, design, and application." Journal of microbiological methods 43, no. 1 (2000): 3-31.

[40] Al-Smadi, Mohammad, Omar Qawasmeh, Mahmoud Al-Ayyoub, Yaser Jararweh, and Brij Gupta. "Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews." Journal of computational science 27 (2018): 386-393.

[41] Parmar, Aakash, Rakesh Katariya, and Vatsal Patel. "A review on random forest: An ensemble classifier." In International conference on intelligent data communication technologies and internet of things (ICICI) 2018, pp. 758-763. Springer International Publishing, 2019.

[42] Rokach, Lior, and Oded Maimon. "Decision trees." Data mining and knowledge discovery handbook (2005): 165-192.

[43] Martinez, Angel R. "Part-of-speech tagging." Wiley Interdisciplinary Reviews: Computational Statistics 4, no. 1 (2012): 107-113.

[44] Cunningham, Padraig, and Sarah Jane Delany. "K-nearest neighbour classifiers-a tutorial." ACM computing surveys (CSUR) 54, no. 6 (2021): 1-25.

[45] Michelucci, Umberto. "An introduction to autoencoders." arXiv preprint arXiv:2201.03898 (2022).

[46] Montúfar, Guido. "Restricted boltzmann machines: Introduction and review." In Information Geometry and Its Applications: On the Occasion of Shun-ichi Amari's 80th Birthday, IGAIA IV Liblice, Czech Republic, June 2016, pp. 75-115. Springer International Publishing, 2018.

[47] Buttle, Francis, and Stan Maklan. Customer relationship management: concepts and technologies. Routledge, 2019.

[48] Madinah, Saudia Arabia. "Emotion detection through facial feature recognition." International Journal of Multimedia and Ubiquitous Engineering 12, no. 11 (2017): 21-30.

[49] Allen, James. Natural language understanding. Benjamin-Cummings Publishing Co., Inc., 1995.

[50] McDonald, David D. "Natural language generation." Handbook of natural language processing 2 (2010): 121-144.

[51] Alanazi, Sarah Saad, Nazar Elfadil, Mutsam Jarajreh, and Saad Algarni. "Question answering systems: a systematic literature review." International Journal of Advanced Computer Science and Applications 12, no. 3 (2021).

[52] Praveen, Shagufta, and Umesh Chandra. "Influence of structured, semi-structured, unstructured data on various data models." International Journal of Scientific & Engineering Research 8, no. 12 (2017): 67-69.

[53] Rose, Stuart, Dave Engel, Nick Cramer, and Wendy Cowley. "Automatic keyword extraction from individual documents." Text mining: applications and theory (2010): 1-20.

[54] Klink, Stefan, Koichi Kise, Andreas Dengel, Markus Junker, and Stefan Agne. "Document information retrieval." Digital Document Processing: Major Directions and Recent Advances (2007): 351-378.

[55] Bauskar, Shubham, Vijay Badole, Prajal Jain, and Meenu Chawla. "Natural language processing based hybrid model for detecting fake news using content-based features and social features." International Journal of Information Engineering and Electronic Business 11, no. 4 (2019): 1-10.

[56] Jakkula, Vikramaditya. "Tutorial on support vector machine (svm)." School of EECS, Washington State University 37, no. 2.5 (2006): 3.

[57] Parmar, Aakash, Rakesh Katariya, and Vatsal Patel. "A review on random forest: An ensemble classifier." In International conference on intelligent data communication technologies and internet of things (ICICI) 2018, pp. 758-763. Springer International Publishing, 2019

[58] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794. 2016

[59] Keller, James M., Michael R. Gray, and James A. Givens. "A fuzzy k-nearest neighbor algorithm." IEEE transactions on systems, man, and cybernetics 4 (1985): 580-585

[60] Zheng, Alice, and Amanda Casari. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc.", 2018.

[61] Wu, Lei, Steven CH Hoi, and Nenghai Yu. "Semantics-preserving bag-of-words models and applications." IEEE Transactions on Image Processing 19, no. 7 (2010): 1908-1920

[62] Mishra, Mridul K., and Jaydeep Viradiya. "Survey of sentence embedding methods." International Journal of Applied Science and Computations 6, no. 3 (2019): 592-592.

[63] Selva Birunda, S., and R. Kanniga Devi. "A review on word embedding techniques for text classification." Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020 (2021): 267-281.

[64] Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. Procedia Computer Science, 157, 160–167. https://doi.org/10.1016/j.procs.2019.08.153

[65] Rong, Xin. "word2vec parameter learning explained." arXiv preprint arXiv:1411.2738 (2014).

[66] Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. "A closer look at skip-gram modelling." In LREC, vol. 6, pp. 1222-1225. 2006

[67] Liu, Bing. "Text sentiment analysis based on CBOW model and deep learning in big data environment." Journal of ambient intelligence and humanized computing 11, no. 2 (2020): 451-458

[68] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014

[69] Yao, Tengjun, Zhengang Zhai, and Bingtao Gao. "Text classification model based on fasttext." In 2020 IEEE International conference on artificial intelligence and information systems (ICAIIS), pp. 154-157. IEEE, 2020.

[70] Ruchansky, Natali, Sungyong Seo, and Yan Liu. "Csi: A hybrid deep model for fake news detection." In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 797-806. 2017

[71] Conroy, Nadia K., Victoria L. Rubin, and Yimin Chen. "Automatic deception detection: Methods for finding fake news." Proceedings of the association for information science and technology 52, no. 1 (2015): 1-4..

[72] Granik, Mykhailo, and Volodymyr Mesyura. "Fake news detection using naive Bayes classifier." In 2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON), pp. 900-903. IEEE, 2017.

[73] Bourgonje, Peter, Julian Moreno Schneider, and Georg Rehm. "From clickbait to fake news detection: an approach based on detecting the stance of headlines to

articles." In Proceedings of the 2017 EMNLP workshop: natural language processing meets journalism, pp. 84-89. 2017.

[74] Gilda, Shlok. "Notice of Violation of IEEE Publication Principles: Evaluating machine learning algorithms for fake news detection." In 2017 IEEE 15th student conference on research and development (SCOReD), pp. 110-115. IEEE, 2017.

[75] Alasadi, Suad A., and Wesam S. Bhaya. "Review of data preprocessing techniques in data mining." Journal of Engineering and Applied Sciences 12, no. 16 (2017): 4102-4107.

[76] Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches." IEEE Data Eng. Bull. 23, no. 4 (2000): 3-13.

[77] Lenzerini, Maurizio. "Data integration: A theoretical perspective." In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 233-246. 2002.

[78] Lei, Shang. "A feature selection method based on information gain and genetic algorithm." In 2012 international conference on computer science and electronics engineering, vol. 2, pp. 355-358. IEEE, 2012.

[79] Celikyilmaz, Asli, Dilek Hakkani-Tür, and Junlan Feng. "Probabilistic model-based sentiment analysis of twitter messages." In 2010 IEEE Spoken Language Technology Workshop, pp. 79-84. IEEE, 2010.

[80] Hassan, Hebatallah A. Mohamed, Giuseppe Sansonetti, Fabio Gasparetti, Alessandro Micarelli, and Joeran Beel. "Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation?." In RecSys (Late-Breaking Results), pp. 6-10. 2019.

[81] Rohera, Dhiren, Harshal Shethna, Keyur Patel, Urvish Thakker, Sudeep Tanwar, Rajesh Gupta, Wei-Chiang Hong, and Ravi Sharma. "A taxonomy of fake news classification techniques: Survey and implementation aspects." IEEE Access 10 (2022): 30367-30394.

[82] Dey, Sanjay, Sarhan Wasif, Dhiman Sikder Tonmoy, Subrina Sultana, Jayjeet Sarkar, and Monisha Dey. "A comparative study of support vector machine and Naive Bayes classifier for sentiment analysis on Amazon product reviews." In

2020 International Conference on Contemporary Computing and Applications (IC3A), pp. 217-220. IEEE, 2020.

[83] Blackfire Technology, “ISOT Dataset Information” https://www.impactcybertrust.org/dataset_view?idDataset=952

[84] Kaggle Organization, “ Fake News Dataset Information” https://www.kaggle.com/datasets/jillanisofttech/fake-or-real-news