Encrypting sentiments: a study on integrating encryption module with NLP pipeline to analyze emotions while ensuring security

by

Sara Jerin Prithila
24241137
Kohinoor Sultana Elora
21101147
Al Rafi Ahmed
21101092
Md. Shamsul Rahat Chy
24341123

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sara Jerin Prithila
24241137

Kohinoor Sultana Elora
21101147

Al Rafi Ahmed
21101092

Md. Shamsul Rahat Chy
24341123

Approval

The thesis titled "Encrypting sentiments: a study on integrating encryption module with NLP pipeline to analyze emotions while ensuring security" submitted by

1. Sara Jerin Prithila (24241137)
2. Kohinoor Sultana Elora (21101147)
3. Al Rafi Ahmed (21101092)
4. Md. Shamsul Rahat Chy (24341123)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on October, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Md Faisal Ahmed

Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

This research delves into the critical role of encryption in Natural Language Processing (NLP), emphasizing its significance as an emergency text platform, akin to a text-based emergency broadcast service. The study analyzes sentiments such as sadness, neutrality, worry, love, surprise, etc., utilizing a standard NLP pipeline for sentiment analysis. Additionally, it compares the results with an enhanced approach that incorporates an encryption module, aiming to quantify potential data loss in the latter scenario and highlighting the trade-offs between data protection and sentiment analysis accuracy in NLP. Addressing the prevalent absence of security components in existing NLP pipelines, this research introduces encryption to enhance security. This academic pursuit sheds light on the nuanced relationship between data protection and sentiment analysis accuracy in the context of NLP, providing valuable insights to guide the refinement of more resilient emergency text-based services while creating a cross-section between the NLP pipeline and Encryption Module.

Keywords: Natural Language Processing (NLP); Machine Learning; Encryption; Substitution Cipher; Polygraphic Substitution; Embedding; Word2Vec; RNN; LSTM; GRU;

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language processing (NLP) is a machine-learning technology that gives computers the ability to interpret, manipulate, and comprehend human language. It provides a crucial link between computer comprehension and human communication, enabling more fluid interactions. On the other hand, during data transmission and storage, encryption is necessary to protect data security and privacy. It's a fundamental component of cybersecurity in a world where cyber threats and data breaches are constants. It entails encrypting information so that only authorized parties can access it.

1.1 Motivation

Natural language processing (NLP) is a machine-learning technology that gives computers the ability to interpret, manipulate, and comprehend human language. It provides a crucial link between computer comprehension and human communication, enabling more fluid interactions. Different dialects, slang terms, tones, and grammatical errors that arise in ordinary, everyday conversation can all be resolved using NLP. Companies use it for several automated tasks, such as processing, analyzing, and archiving large documents, analyzing customer feedback, running chatbots for automated customer service, answering who-what-when-where questions as well as classifying and extracting text. To process human language, this language processing integrates deep learning models, machine learning, and computational linguistics. The study of interpreting and creating human language models with computers and software tools is known as computational linguistics. Researchers build frameworks to assist machines in comprehending conversational human language using computational linguistics techniques like syntactic and semantic analysis. Machine learning is a method that increases a computer's efficiency by teaching it using example data. Sarcasm, metaphors, differences in sentence structure, as well as grammar and usage exceptions that take years to acquire, are just a few of the characteristics of human language. Machine learning techniques are used by programmers to train natural language processing (NLP) software to recognize and comprehend these qualities right away. These days, sentiment analysis is a use of natural language processing (NLP) technology to teach computer programs to comprehend text in a manner akin to that of a human. It evaluates digital text to identify whether a message is positive, negative, or neutral in terms of emotion. Usually, the analysis goes through multiple phases before yielding the ultimate outcome. On the other hand, sentiment analysis entails examining the writer's psychological state at the time of writing. Sentiment analysis goes beyond simple categorization, making emotional detection a more intricate

field. This NLP pipeline, with the help of models, helps identify different feelings, like happiness, rage, sorrow, and regret, based on the individual's selection of words. It is through the expression of emotions and sentiments through this language that help in understanding what someone is going through, what someone is feeling, what situation they are in, etc. In this case, how someone expresses their feelings and emotions becomes very important because it helps us understand not just what they are saying but also their underlying emotions and the situational context.

In light of the constant progress in technology, the modern world is changing quickly, changing the way one collaborates, works, and lives. Technology has the potential to be a very positive thing. Technological advancements such as blockchain, quantum computing, and artificial intelligence are revolutionizing various industries by facilitating automation and improving productivity. Globalization and remote collaboration have been fueled by digital connectivity, which is enabled by the Internet and 5G. This has increased access to information. Technology is a key economic engine for those who think that innovation and the potential for creative destruction can promote economic progress and improve quality of life (Schumpeter 1942). It may, however, also be an extremely oppressive and terrifying tool. Along with the issues that come with technical advancement are cybersecurity dangers and ethical concerns about privacy and AI usage. Encryption is necessary in all spheres including healthcare, financial service, postal service, mental health service, and e-commerce [28]. Clients have to input their confidential data to access these over the internet and service providers might need to analyze the texts. Societies need to adjust to the effects of technology's ongoing evolution while maintaining sustainable and ethical development.

1.2 Challenges

Machines still struggle to understand human language, even with advances in natural language processing (NLP) technologies. They might not understand the subtler aspects of human communication. For example, sentiment analysis in words containing sarcasm is a very challenging task for a machine. Secondly, negation is the process of expressing a reversal of meaning in a statement by using terms that are negative. Sentiment analysis algorithms may find it challenging to accurately read these kinds of statements, especially if the negation occurs between two sentences. Again, when a sentence expresses more than one sentiment, it is said to be multipolar. The software finds it more challenging to understand the underlying sentiment. Challenges also include adapting to different topics, industries, or dialects. It is confirmed that the model can manage these difficulties by testing on sentiment analysis, as similar problems are frequently encountered in more general NLP applications.

Cybersecurity is crucial to the technological industry, yet safeguarding data has emerged as one of the major problems of the modern day. Billions of people around the world rely on technology for data-related tasks. Consequently, the issues associated with cyber security are growing daily. The main reason can be said to be that it is convenient and less risky than physical attacks with the expansion of cyberspace. Modern commercial organizations find themselves at the vanguard of innovation in the ever-evolving corners, but they also find themselves in the crosshairs of persistent cyber-attacks. It is thus essential to ensure confidentiality, integrity, and authenticity.

Robust encryption models need to use robust encryption algorithms and protocols along with safe key management to combat these threats. They have to prevent both external and internal manipulation and illegal decryption. The necessity for thorough encryption techniques to protect sensitive data is highlighted by the vulnerability of short substituted words to brute force decryption.

1.3 Importance

With the advancement of information technology and the rise in internet usage, it's getting harder to communicate one's emotions over the internet without feeling exposed to dangers. **This paper explores a specific case where the use of NLP can be highly beneficial, particularly in emergency situations where individuals may struggle to ask for help, while also ensuring confidentiality and integrity during communication.** Although this is still a relatively untapped topic, further research has not yet been done on the cross-section of encryption modules and natural language processing, with very few exceptions. As an illustration, SecureNLP [29] is a cutting-edge privacy-preserving solution that effectively combines NLP services while protecting user privacy and data utility. SecureNLP, employing a recurrent neural network (RNN)-based sequence-to-sequence (seq2seq) model with attention mechanisms, proposes efficient distributed protocols using secure multi-party computing (MPC) for non-linear functions like sigmoid and tanh. The suggested methods protect sequence-to-sequence transformations (PrivSEQ2SEQ) and long short-term memory networks (PrivLSTM) from semi-honest adversaries. Again, the use of chatbots in a variety of industries has increased due to growing privacy concerns. Although earlier studies have mostly concentrated on improving NLP accuracy, the researchers have suggested methods that include privacy-preserving features. In PrivFT [27], the researcher has approached to classify text and at the same time to protect user inputs using fully homomorphic encryption (FHE) safeguarding user information and their environments becomes paramount to ensure secure communication and storage. It is imperative to develop a system that not only protects user privacy but also guarantees that data analysis upholds privacy and NLP standards.

This study addresses safety concerns in sensitive settings when people might find it difficult to discreetly ask for help due to safety concerns. This research suggests creating a secure pipeline that analyses messages to identify the intended emotion in addition to shielding them from vulnerabilities in security. It becomes imperative to apply sentiment analysis to the text message in order to prioritize and identify important information. Messages are encrypted during transmission and decoded when they reach the recipient's end to increase security. After decrypting the data, the computer does thorough syntactic and semantic analyses to transform unstructured material into a comprehensible format. The main goals of this research are emotional insight and privacy, both of which are ensured by this approach. Previous studies focused primarily on enhancing NLP accuracy or incorporating privacy-preserving features. This distinguishes it from conventional implementations, which might not adequately tackle the combined issues of privacy and sentiment analysis in high-stakes settings.

1.4 Research Problem

This research represents a cross-section of the NLP Pipeline and Encryption Module, aiming to develop an efficient method for detecting and analyzing sentiments in text while ensuring security and assessing whether sentiments remain intact even after encryption. In this age of technology, with the widespread availability of the internet, over the years communication has become easier than ever. Ensuring privacy and creating a secure channel for communication has become a major concern as well. People tend to express their opinions through social media posts and communicate over the internet by any means expressing their emotions. Sometimes these posts and messages contain sensible information or expose emotional vulnerability which could be risky and at the same time security and confidentiality are required.

In case of emergency or to share vulnerable information, communication channels must be secured. Now, if the receiver could get a hint of the message's emotional phase or the sentiment, it could help to take necessary action. Moreover, nowadays government authorities and many non-government organizations (NGOs) have emergency services. Most of the time, they receive a huge amount of requests i.e.broadcast messages at a time. It would be beneficial to take prompt action if the message could be classified based on the emotional intensity and sentiment analysis that the texts' contain. Even then the system could raise an alarm for any danger forthcoming by automatic analysis keeping the message secure without exposing it to the wrong party.

Furthermore, taking into consideration the risk factors of cybersecurity, intruders, and hackers always seek to access users' data including messages, and vulnerable information. Therefore, the whole process needs to be secured and confidential as well and for better efficiency of desired work, emotion detection or sentiment analysis is required.

Besides, this research could unfold other fields of security and emotion detection. To grow businesses, social media marketing is a common trend and arguably one of the effective methods. By emotion detection and sentiment analysis, while keeping the users' information safe, companies can ensure effective systems to automatically show proper ads.

In Badawi et al's (2020) [27] paper, the researcher has approached to classify text and at the same time to protect user inputs. Nowadays, automation is taking place in all spheres including healthcare, financial service, postal service, mental health service, and e-commerce. Clients have to input their confidential data to access these over the internet and service providers might need to analyze the texts.

Not only that, it often comes into the limelight that depressive and suicidal people tend to show hints through their daily opinions, social media posts, and even conversations. If there is any system that can detect symptoms by analyzing sentiments and detecting emotion keeping all the information secure and private. It would be more appropriate if it could be done from the encrypted information.

In the paper by Feng et al. (2020) [29], the issue of privacy persevering while performing NLP or ML tasks has been addressed. At present, most of the applications take users'

inputs and interests and then try to show suggestions or products based on that. Also, it is a must to keep users' inputs private and secure.

Not a lot of research has been done on figuring out if NLP data's text and semantical properties, after being encrypted and taken to an encrypted space, remain the same. In order to solve the issue, this research will concentrate on identifying emotion, evaluating sentiment, assuring security, and presenting the reality of encrypted sentiment preservation.

1.5 Research Objectives

In light of this, the study aims to achieve the following distinct objectives:

- To effectively use the encryption module to construct a cross-section with the NLP pipeline.
- To use encryption techniques to protect confidentiality and privacy while utilizing the NLP pipeline to identify emotions and analyze sentiments.
- To explore how emotion detection models for encrypted text operates.
- To assess how well NLP networks and encrypted text function together.
- To evaluate the semantical property preservation of words after being taken to an encrypted space.
- To develop an effective model for emotion detection and sentiment analysis from encrypted text or decrypted text which is convenient.
- To create a system that will aid in handling emergency situations and ensure the utmost security for victim data.
- To understand the drawbacks and find out the risk for future works.

Chapter 2

Literature Review

For our literature review, we've chosen a wide range of papers that provide a comprehensive understanding of sentiment analysis (SA), text-based emotion detection (ED), and privacy-preserving natural language processing (NLP) systems. In order to delve into the topic of Natural Language Processing (NLP), we carefully chose relevant papers from reputable websites such as ResearchGate and IEEE. The citation analysis approach was essential in helping us find significant publications and recent changes in NLP. We sifted through a large number of encryption-related publications, paying attention to include both seminal works and modern contributions. Our supervisors' guidance and our efforts helped us understand the fundamentals of both NLP and encryption. This gives a comprehension of these areas.

The literature review provides a comprehensive understanding of sentiment analysis (SA), text-based emotion detection (ED), and privacy-preserving natural language processing (NLP) systems. The survey conducted by Wankhade et al. provides an overview of the many approaches used in sentiment analysis (SA), such as hybrid approaches, machine learning, lexicon-based approaches, aspect-based SA (ABSA), multimodal SA (MSA), and transfer learning. Sentiment analysis uses text mining and natural language processing to find and extract subjective information from the text. This paper explores sentiment analysis and the different approaches, problems, tools, and algorithms used in it. Lexicon-based techniques have limitations in terms of domain specificity and rely on predefined token scores. Lexicon-based methods' sub-approaches, like Corpus-Based and Statistical Approaches, are designed to get beyond these restrictions. Supervised and Lexicon-based Unsupervised Learning are the two main methods of machine learning used to categorise feelings. The Hybrid Approach emphasises the significant usage of sentiment lexicons by deliberately combining the two techniques. ABSA stands out as a unique approach that detects features, classifies sentiment, and aggregates the output, whilst Transfer Learning and MSA provide flexibility in a variety of contexts by utilizing trained models and combining audio and visual input. These techniques have special advantages in addressing the nature of sentiment in various data sources as sentiment analysis develops. It has a wide range of uses, from improving corporate intelligence to developing entertainment and healthcare. In the business world, SA develops creative marketing ideas, tests customer input, and improves goods. Sentiment analysis is used in healthcare to improve services by assessing drug reactions, tracking epidemics, and gauging patient moods. Entertainment gains from analyzing viewer responses to films and shows, aiding content popularity. Furthermore, sentiment analysis helps anticipate stock

prices by assessing news, spotting market patterns, and guiding financial choices. Despite these advancements, challenges persist in SA, particularly in tasks like sarcasm detection and interpreting informal writing styles featuring acronyms, emojis, and shortcuts. [41]

Sosea and Caragea (2021) [37] have proposed Emotion Masked Language Modeling (eMLM) which is a BERT (Bidirectional Encoder representations from transformers) that targets sentiment-specific biases at pre-training without any extra computational cost. Generally in BERT tokens that carry strong emotional words belonging to a lexicon have higher masking probabilities. For this, they have lowered other masking probabilities, which have less emotionally rich words to keep the sum of probabilities constant, in which a total of 15% of words are masked. A masking probability of 0.5 worked best for their model. The model has been trained and tested using multiple data sets, including Stanford Sentiment Treeback SST, Go emotions, and CancerEMO. For lexicon, EmoLex has been used. These datasets include 27 emotion categories, sentence labels, and multi-level labels, providing a comprehensive range of sentiment-level data. Sentiment and emotion words were considered separately eMLM (E) masked emotion-revealing words, and eMLM (S) masked sentiment-revealing words, in which eMLM (E) outperformed suggesting that emotion words produce better representations for the task. eMLM produces high-quality contextualized embeddings for the task of detecting sentiment analysis and emotion. Also, for future work, they aim to determine whether the models' performance will be impacted by involving emotion intensity, with the masking probability.

The paper by Saju et al. (2020) [34] aimed to give a complete assessment of sentiment analysis, including its kinds, techniques, contemporary applications, tools, and APIs. Binju et al. (2020) did a literature evaluation of several academic publications, conference proceedings, and books on sentiment analysis. They looked at several forms of sentiment analysis, such as rule-based and automated techniques, as well as machine learning algorithms for sentiment categorization. Binju et al. (2020) presented a thorough overview of sentiment analysis, emphasizing advances in machine learning and natural language processing approaches. They also covered the different tools and APIs available for sentiment analysis, such as open-source libraries and commercial applications. However, this research entails investigating the ethical implications of sentiment analysis, such as privacy problems and prejudice, as well as tackling the challenges of sarcasm and irony in sentiment categorization. Sentiment analysis research in the future will examine the moral ramifications, deal with irony and sarcasm's difficulties, and expand its use to other fields.

Acheampong et al.'s (2021) paper [35] titled 'A transformer-based approach to irony and sarcasm detection' has found a better algorithmic approach to detect figurative language classes. Using figurative language (FL) has been a common trend in online social media platforms. Figurative language mainly consists of three types of expressions such as sarcasm, irony, and metaphor. Detecting or identifying norms in short texts' figurative language is quite challenging and still an unresolved issue for the natural language processing field. Nowadays, people tend to express their emotions on social media in social media. Natural language processing (NLP) methodologies are effective for normal text but the performance drops when it comes to figurative language. Hence, this newly modified advanced deep learning method proposed by the authors could be a new horizon to tackle the problem of figurative language in NLP. There have been many experiments

carried out to identify the small text of figurative forms from multiple approaches based on different parameters. For better prediction, it needs to keep track of textual patterns, contextual embeddings, and long sequences and feed into the algorithm. One of the popular NLP methods, Transformer methodologies are popular for machine translation, and text summarization, and with time BERT, XLNet, and RoBERTa models have been developed. Pre-trained networks are beneficial and pre-trained RoBERTa can efficiently map words onto a rich embedding, long sequence dependencies. On the other hand, RCNN can capture temporal relations and dominant words. Therefore, the authors of this research have proposed a new architecture, a combination of RCNN and RoBERTa. The proposed method consists of a pretrained transformer, RoBERTa (with 12 RoBERTa attention heads) followed by a bidirectional LSTM layer including 12 LSTM units and 0.1 LSTM dropout. After concatenation of RoBERTa and LSTM, a pooling layer then finally passes into a fully connected softmax layer. The proposed method's performance has been compared with other popular NLP architectures using three different datasets. For all three datasets, the proposed RCNN-RoBERTa outforms in all metrics of other network models. A point to be noticed is that, using only RoBERTa architecture, the performance score is second highest for most cases but integrating RCNN with RoBERTa clearly improved the performance result in detecting figurative language forms. However, further improvement and research could be carried out for better accuracy and to overcome the remaining obstacles of figurative language.

The study done by Poria et al. (2023) [42] covers both past and future applications and directions of sentiment analysis. The authors discussed recent sentiment analysis trends and problems. There are different types of sentiment analysis, the process of document-level sentiment analysis involves inferring the overall opinion of a document, while sentence-level sentiment analysis focuses on individual sentences, and phrase-level sentiment analysis focuses on a phrase within a sentence. The challenge with phrase-level sentiment analysis is that words in a lexicon can change in and out of context. Rule-based sentiment analysis focuses on learning through heuristics and rules, where lexicons are utilized, here lexicons are a type of dictionary that contains sentiment annotations and helps to understand the sentiment polarity of a word. CNN (Convolutional neural networks) and RNN (Recurrent neural networks) are widely used for feature extraction in the field of deep learning with the aid of word embeddings. In the case of sentiment analysis, the words around a sentence play a great role in conveying contextual meaning. Individuals may sarcastically convey their emotions, for instance, if person A says, "The order has been canceled." and in response to this person B says, "This is great!". Here person B is being sarcastic, even though the word "great" should be considered as a positive emotion but the word "great" here has a negative polarity, depending on the speaker's character and context. So, the authors suggested that to improve sentiment analysis results, speaker profiling and previous utterances in a conversation are crucial and only to be dependent on deep learning frameworks and word embeddings that lack these types of background information. Dealing with bias in sentiment analysis is crucial, as frequent use of meta-information such as race and geographic cues can lead to geographical biases. For example, opinions about "good traffic" and "cheap phone" could differ between an Indian and an American individual, so applying data originating from a different geographical location can result in the model generating demographic bias. The Equity Evaluation Corpus (EEC) can be used to evaluate racial biases. Substituting a word in a text for alternative cases and then minimizing embedding spaces or the

difference in prediction between the altered and the previous text can be an approach for de-biasing. For future work, the author suggested analyzing an individual's personality by examining their past tweets as a possible future area of study. Studying the inter-aspect dependency of a word is crucial for improving the performance of Aspect-level sentiment analysis. Moreover, including contextual sentiment-bearing phrases, which is a type of phrases that initially sound neutral but whose meaning can change when combined with other words, in a dataset can be an excellent contribution to the research.

The work of M. M. Tadesse et al. (2019) [24] highlights the elusive nature of user tone, which is a serious concern. In order to improve the accuracy of depression identification, this paper aims to: 1)Examine the relationship between depression and language use; 2)Design three LIWC features (linguistic dimensions, psychological processes, and personal concerns); 3)Evaluate the efficacy of N-grams probabilities, LIWC, and LDA as single features; and 4) Show the predictive strength of both individual and combined features using novel classification approaches. Depressed users were found to focus more on the present and future, speak more about themselves, and experience more negative feelings. Once more, it is discovered that if LIWC is included in tool design, it can effectively support data detection models. Combining LIWC, LDA, and bigrams is an effective way to diagnose depression; the MLP neural network model outperforms SVM, LR, RF, and AdaBoost, with 91% accuracy and 0.93 F1 score. The choices and combinations of features have a big influence on performance. The MLP classifier is a prime example of the resilience of integrated characteristics; it achieved an impressive 91% accuracy and 0.93 F1 score, indicating greater efficacy in diagnosing sadness in Reddit social media within the parameters of this study. When comparing LIWC with LDA features, LIWC performed better than topic models created by LDA. Even if the experiment shows respectably high methodological performance, the absolute metric numbers highlight how difficult this endeavour is, calling for more research. This experiment could contribute to establishing a foundation for innovative mechanisms in diverse healthcare domains, facilitating the estimation of depression and related variables.

COVID-19 has been a major concern among the mass people and greatly affected mental health. Here, in this research paper of Imran et al. (2020) [31] have tried to analyze and identify emotion and cross-cultural polarity using sentiment analysis and deep learning by collecting COVID-19-related tweets. The goal of this research is to infer and analyze how citizens from different cultures react or respond to the coronavirus and their sentiments on the actions taken by different countries. This research is limited to detecting six primary emotions such as joy, surprise, sadness, fear, anger, and disgust. Prior to classifying emotions, the authors have initially distinguished these six emotional categories into two segments based on sentiment polarity which are positive tweets and negative tweets. For datasets, the authors have collected tweets to prepare their own dataset using coronavirus-related hashtags and used two other Kaggle datasets namely Sentiment140 and Emotional Tweets dataset. The authors used the Kaggle datasets for training their proposed model. The proposed network architecture is a multi-layer LSTM assessment model. Before feeding into the model, authors followed the PRISMA approach to arrange COVID-related tweets and also mentions, colons, stop words, emoticons, and punctuations were removed. Moreover, tokenization had been applied for cleaned tweets. They trained the model in three steps, at first using the Sentiment140 dataset for polarity then the Emotional Tweet dataset for positive polarity and negative polarity respectively. Fasttext

word embeddings with LSTM have been used to classify tweets' polarity and achieved an accuracy score of 82.4%. While using the emotional tweet dataset for training actual emotions, the authors used the LSTM network integrated with pre-trained Glove Twitter which outperformed other models both in accuracy score and F1 score. The accuracy scores for positive polarity emotions and negative polarity emotions are respectively 81.9% and 69.9%. The authors validated the model's performance with classified emoticons and also showed Pearson correlation to analyze the sentiments and emotion trends among the countries. The findings of this research are a significant contribution to analyzing tweets using NLP pipelines. However, this research has limitations as the training was carried out for just the English language and at 280 characters in length. Further research in the future is needed to improve the lackings.

[2] The intersection of natural language processing (NLP) and information assurance and security (IAS) is studied in the research done by Atallah et al.(2000) . It explores the ways in which natural languages encode meaning in intricate and frequently unforeseen ways, and how these ways may be used to improve information security. The paper lists four possible uses for it. (i) leveraging machine translation systems for further text message encryption; (ii) watermarking in natural language; (iii) downgrading or sanitising classified material in networks; and (iv) using automatically generated humorous jingles to aid in memorising random passwords. The second and third applications are still in the proof-of-concept phase, although the first and fourth applications have seen some partial implementation. The creative potential of NLP, especially ontological semantics, for IAS is highlighted in this work. The co-authors' contributions to the study include theoretical and empirical components, as well as software for password memorising, meaning analysis, watermarking, and knowledge representation. The authors made the decision to publish their preliminary results in order to solicit input and cooperation from the information security community. To aid IAS professionals in understanding these subjects and investigating their applications in information security, the study offers appendices on ontological semantics and natural language processing. According to the research, more cooperation will lead to more NLP applications, improving the security and assurance of computer information.

Similarly, the KR approach and the Lexical Affinity (LA) method are introduced in the work by Acheampong et al. (2020) [26] Emotion dictionaries or lexicons with keywords like "happy" or "angry" are used in the KR approach. By giving words associated with random emotions probabilistic affinities, the Lexical Affinity (LA) approach expands on KR. Many datasets, including ISEAR, SemEval, EMOBANK, WASSA-2017 EmoInt, Alm's Affect, Daily Dialogue, AMAN's Emotion, Grounded Emotion, Emotion-Stimulus, Crowdsourcing, MELD, Emotion lines, and Smile, are used in emotion analysis research. The results of this study point to a significant research gap in text-based Emotion Detection (ED) for vital, life-saving uses. Certain domains, such as detecting criminal activity via victim message analysis to detect potential threats and evaluating depression levels in patient texts to provide prompt assistance, are yet not fully investigated. Researching these life-saving applications and related fields would not only advance the field of text-based ED but also greatly address important social demands, especially considering the large amount of text data that is generated every day.

Mahendran et al(2021) [36] researched the connection and relationship between (NLP)

natural language processing and privacy. They tried to find the technique in NLP to identify the research that is about privacy. Additionally, evaluated them based on the safety and information that's about privacy threats. The researchers discovered sensitive private data through social media, applications, medical records and multiple software programs. Here, they figure out that NLP-based solutions can be crucial for data privacy and maintaining that privacy. There were four areas that were concerning. They are software privacy and security, online social networks (OSN), PHI encryption and de-identification in healthcare, and methods of deidentification. Their study explored protecting patients' medical data from unauthorized access. However, their focus was to spread awareness among people's hreats towards personal information. The researchers categorized the privacy that uses NLP into four categories. They inducted NLP is crucial and significant for data privacy problems and also ensuring the privacy of the data. Analyzing previous studies helps the research avoid NLP sector invasions. Nonetheless, insufficient coverage of particular applications or case studies might make a lacuna in providing viable insights into implementing privacy-preserving methods in real-life scenarios. In the subsequent study, Mahendran et al. (2021) might delve deeper into cases or realistic instances that apply privacy-conserving tactics expounded in the survey. Moreover, investigating what morality has to say about employing NLP concerning privacy protection and handling scaling issues which might arise when employing broader applications would serve entrepreneurs well in other areas.

The research done by Jing et al.(2012) [6] introduces a new text encryption approach that is similar to text watermarking and makes use of natural language processing (NLP) techniques. Syntactic, semantic, and synonym substitution are the three main linguistic alterations that are incorporated into the algorithm. Using resources like WordNet, synonym substitution entails swapping out terms with their synonyms while preserving the content of the phrase. Semantic transformations employ coreference notions to preserve meaning through strategies like coreferent pruning, whereas syntactic transformations only modify the sentence structure with little effect on meaning. The original text is converted to binary, it is embedded in a cover text by modifying its semantic units, and a dependency tree structure is employed as part of the encryption process. Using a secret key, every node in the tree is labelled and transformed to binary based on a quadratic residue check. Sentences are scored using a hash function, and node sequences are sorted by length. Based on this ranking, the original text is then placed into the cover text. This method shows how text watermarking techniques can be modified to improve text encryption, providing a high-level information security solution that maintains the original text's meaning using cutting-edge NLP algorithms. Although this technique is still in its infancy, it has the potential to produce strong text encryption.

Mishra et al. 's (2024) [47] research addresses privacy and security concerns in deep neural language models used in AI applications. These models, which are frequently installed on internet-accessible servers, run the danger of data storage-related privacy violations and user input interception. The authors suggest a novel approach to modify and improve transformer-based language models on passkey-encrypted text in order to address these problems. This entails transforming the tokenizer and token embeddings irreversibly in order to avoid reverse engineering and enable secure inference on encrypted inputs. The process involves fine-tuning on encrypted datasets after pre-trained models such as BERT and RoBERTa are adapted without additional pre-training. By using this

method, performance parity between the models and their unencrypted counterparts is guaranteed. In-depth analyses of this approach on benchmark datasets for text classification and sequence labelling tasks are used in the research to show how effective it is. Novel transformer model adaption strategies, improved text security via encryption, and empirical validation demonstrating that the approach preserves model performance are among the major advances. In an effort to strengthen user privacy and security in useful AI applications, the authors intend to expand this method to generative models and investigate cutting-edge encryption techniques.

In order to improve sentiment categorization in Twitter data, Kanakaraj and Guddeti (2015) [14] suggested an ensemble classifier-based Natural Language Processing (NLP) method. The idea behind using NLP approaches was to improve sentiment categorization accuracy. Using the user search key as a term, the researchers used the Twitter API 1.1 to collect data from Twitter. After that, they eliminated stop words and cleansed the data for linguistic errors like noise and repeated letters or phrases. A feature vector was created using keywords, senses, and associated synsets. Different subsets of the training data were used to train multiple models, and ensemble approaches were used for classification. The proposed approach included modules for data collection, data processing, training and classification, and classification output, allowing for the incorporation of NLP techniques into the sentiment analysis process. This enabled a more accurate interpretation of the mood suggested by the Twitter data. Kanakaraj and Guddeti's research showed that the semantics-based feature vector combined with an ensemble classifier outperforms the classic bag-of-words strategy with a single machine-learning classifier by 3-5%. The findings highlight the value of combining NLP approaches with ensemble classifiers in sentiment research. However, the suggested solution is distinguished by its novel application of NLP techniques, specifically synsets and word senses, as features in sentiment analysis systems.

In their paper, Gaind et al. (2019) [21] have proposed a different approach for emotion detection instead of identifying the sentiment polarity of social media texts. They have addressed the problem of detection, classification, and quantification of emotion recognition. They have proposed a method that can classify the texts into six emotional categories. The method consists of two different approaches which are extracting textual features using the NLP pipeline and classifying using machine learning algorithms. To test the proposed model, the authors mainly used two datasets. The authors created a Tweets Set by collecting users' tweets and gathered a high-quality quality, accurate bag of words, namely Emotion-words Set (EWS). At first approach, the authors used the Stanford CoreNLP tools to analyze linguistic features for instance recognizing context, finding hits, tokenizing, annotating, and applying negation if necessary. Besides, each of the datasets was preprocessed prior to its use. They removed the unwanted characters, hyperlinks, and hashtags. For the second approach, the authors used machine learning classifiers including SMO and J48. In this phase, by filtering stop words and lowering all words the authors also pre-processed the data. The accuracy score of the SMO classifier was 91.7% and for the J48 classifier, it was around 85.4%. They have made a system for automatic tweet labeling and added a surety factor. However, there are still many spheres left for improvement in future work such as the automation of token collection and making an emotion-specific bag of words.

Chong et al. (2014) [9] analyzed tweets to investigate the application of natural language processing algorithms in sentiment analysis. They conducted their experiment using tweets acquired from Twitter. The tweets were preprocessed to superfluous data and remove noise. The methodology used NLP techniques to identify sentiment expressions for specific situations and identify the polarity of sentiment lexicons. They used feature extraction techniques to extract subject-specific traits and sentiment from each sentiment-bearing lexicon and later on linked the resulting sentiment to specific subjects. Using the feature extraction method, the researchers were able to assess news articles and general websites with an accuracy of 91-93% and online reviews with an accuracy of up to 87%. The authors highlighted the efficiency of NLP techniques in identifying the sentiment of tweets on some topics. However, in the future, the study might focus on researching the use of NLP techniques and sentiment analysis in various fields, including healthcare and politics.

With the aim to enhance the performance of emotion classification Ying et al. (2019) [25] suggested a procedure by incorporating domain knowledge with general knowledge on the language models. This is done using a Twitter-specific preprocessor and by identifying effect-bearing token patterns through a two-step training process. Ekphrasis, which is a Twitter-specific tokenization preprocessing tool, has been used to gain domain information. Special expressions like emoticons, dates, times etc were recognized by this preprocessing tool. Afterwards, Informative token patterns were identified using a convolution network where the output of max-pooling creates a domain-specific representation for a sentence. SemEval-2018 dataset's training set which is multi-labled was used to fine-tune BERT. The trigram model simplifies the learning of complex relationships, including negative and long-distance connections. CNN detector was used as a token pattern detector which will help to add domain-specific knowledge to BERT. Here CNN detector with the help of word2vec learned word embeddings from unlabeled tweets. The CNN detector had the poorest performance because of the simplicity of the tri-gram model, on the other hand, fine-tuning with Twitter data the BERT showed slightly better results than previous state-of-the-art models. The BERT model benefited from the domain knowledge provided by the convolutional neural network CNN detector, making domain knowledge the key to outperforming the state-of-the-art model. Despite being useful and effective indicators, emoticons (':-)') and hashtags (#Christmas) expressions are still not fully utilized by BERT. Therefore, Twitter-specific features improve the accuracy by one percent in emotion classification, moreover, the vanilla BERT does not have domain-specific knowledge. Furthermore, for future work, they assumed introducing elaborate domain knowledge could be a great contribution which may enhance the overall performance.

Naik et al. 's (2018) [19] research paper proposes an online complaint bot to replace traditional written complaint methods, addressing issues like complaint loss and security. The bot prioritises complaints based on keywords and routes them to the relevant authorities using NLP and onion routing to maintain anonymity. Every grievance is given a tracking ID and timestamped. Conventional complaint boxes are vulnerable to theft, alteration, and privacy violations, which deters complainants frequently. On the other hand, users of the planned system can register complaints online using their identify or an anonymous one. Using keywords and tags, complaints are ranked in order of priority, with critical situations such as terrorism marked for quick resolution. Since onion routing protects data security and integrity, it becomes more difficult to identify the complainant.

In order to facilitate safe communication, the system modifies the Tor network. Data is encrypted and routed across several nodes to thwart tracking. The complainants' high level of security and anonymity is preserved by this way. Higher officials can also keep an eye on the status and development of complaints thanks to the complaint bot. All things considered, the suggested approach streamlines the grievance procedure, diminishing information loss and enhancing monitoring, which in turn motivates more individuals to file complaints by protecting their privacy and guaranteeing prompt resolution.

Clinical notes in electronic health records (EHRs) are important for research and healthcare, but because they contain Protected Health Information (PHI), they might be private. The accuracy and efficiency of conventional de-identification techniques, whether manual or automated, are constrained. In order to overcome the difficulties of collaborative research, this paper by Sadat et al. (2019) [23] suggests a unique framework for de-identifying clinical notes from various sources. By using safe thresholding and private set intersection to filter out low-frequency bigrams, the system preserves data utility while guaranteeing privacy, building upon the single-source approach of Li et al. The method improves security during data processing by utilising homomorphic encryption. The usefulness of the system in maintaining data integrity and privacy is demonstrated by experiments carried out on the MIMIC-III dataset. By dividing up computational work among several data owners, the suggested approach guarantees safe communication and effective processing. The framework is ideal for real-world applications since the results show that it can handle big datasets with reasonable computational and communication costs. This study offers a scalable and safe alternative for collaborative research by applying privacy-preserving algorithms to de-identify clinical notes in a dispersed context.

The intersection of NLP and privacy is explored in the paper by Feng et al. (2020) [29]. It introduces SecureNLP, a privacy-preserving system. With strong user privacy and data utility preserved, SecureNLP is a novel privacy-preserving technology that incorporates NLP services effortlessly. The goal of this work is to improve SecureNLP by employing a sequence-to-sequence (seq2seq) model with attention mechanisms that are based on recurrent neural networks (RNNs). It proposes efficient distributed protocols utilizing secure multi-party computing (MPC) for non-linear functions like sigmoid and tanh. The suggested methods protect sequence-to-sequence transformations (PrivSEQ2SEQ) and long short-term memory networks (PrivLSTM) from semi-honest adversaries. For activation functions, the design incorporates multi-party interactive protocols with additive and multiplicative secret sharing. RNN operations are condensed in SecureNLP into PrivSigm and PrivTanh, which are the building blocks for the sigmoid and tanh activation functions. These basic building blocks are then used to design interactive protocols that guarantee privacy in seq2seq using attention models based on RNNs. The protocols provide data privacy while enabling seq2seq transformations for all parties. The suggested method exhibits effectiveness as well as security. PrivSigm and PrivTanh's accuracy closely resembles that of single-party functions, demonstrating its versatility across a range of natural language processing applications that depend on sigmoid and tanh activation functions. It follows that the locally realizable linear operations have no appreciable effects on the performance. However, activation functions provide difficulties for the intended SecureNLP. Certain low-cost, lightweight devices may still incur high computational costs while performing multi-party multiplication. In order to address this, the next research will focus on improving these protocols and investigating lightweight

substitutes such as the elliptic curve cryptosystem. Therefore, a critical topic for further research and development is an optimized version with more effective operations.

In the article [15], the authors Ekta Agrawal et al. (2017) tried to explore a fast and secure way to encryption and decryption of message communication. They aimed to ensure data security while providing effective encryption techniques. They have utilized many encryption techniques to hide the data for security to ensure that the information can not be retrieved easily. They need a specific key to retrieve the message through decryption. The author's methodology was mainly implementing multiple encryption algorithms to safeguard all the messages. They tried to introduce a safe and reliable method to do encryption for ensuring data protection during data transmission and data storage. Ekta Agrawal et al. (2017) achieved a secure and effective encryption-decryption method to process the communication. Through their study, they were able to achieve it. The encryption technique they used successfully concealed the data such as sensitive information that ensured only authorized people could access the data. There are some strengths and weaknesses in their work too. The author's approach provided robust data security and data privacy through encryption, its a strength. The implementation of these encryption techniques enhanced the integrity of their messages. Apart from that there are potential weaknesses too such as managing the encryption keys are complex task as well as ensuring the secure key distribution. However, in the future, Ekta Agrawal et al. will explore the find more advanced encryption techniques for data security. Also, they may investigate methods to streamline the key managing process. It mostly affects the encryption-decryption process. Their future could be in optimizing encryption techniques. Those ming be different for different types of data.

Another similar instance can be noticed in Badawi et al's (2020) [27] paper as it aims to develop an efficient method for text classification while preserving privacy using fully homomorphic encryption (FHE). Their proposed method Private Fast Text (PrivFT) can infer directly from encrypted user inputs and train an effective model based on an encrypted dataset. The research objective of this work is to generate a model on the cloud, protect privacy, and maintain confidentiality of the users' sensitive data input. Here, the proposed fully homomorphic is a kind of encryption scheme that enables the system to compute encrypted data directly without decryption. THE has some limitations too such as high computational overhead and only addition and multiplication on encrypted data are naturally supported. For this research, the authors set their goal to text classification using encrypted data. The authors introduce a shallow neural network for text classification using FHE data. Moreover, to overcome some performance barriers of CPU-based fully homomorphic encryption, the authors have implemented the method (PrivFT) on GPU also. GPU implementation is done by providing a Residual Number System (RNS) variant of the Cheon-Kim-Kim-Song (CKKS) leveled FHE scheme. By applying homomorphic addition and multiplication, data are secured through fully homomorphic encryption. Besides, noise management is a crucial factor in FHE. For noise refreshment, bootstrapping is followed in FHE scheme. For text classification, traditionally LSTM, Bidirectional LSTM, BERT, and Transformer architecture have been used and these are not feasible with fully homomorphic encrypted data when implementing the CKKS variant for GPU. Hence, the authors approached it in a new way called 'fasttext' which consists of a hidden layer and output through a fully connected layer. The authors run their proposed model (PrivFT) on four different datasets and have been able

to achieve near accuracy of traditional NLP networks including XL Net, BERT ITPT, and ULMFit. For the Yelp dataset, the PrivFT accuracy score is 95.41% while XL Net achieves the highest accuracy of 98.45%. Again, for AG and IMDB datasets, XL Net tops in accuracy scores which are 95.51% and 96.21% respectively compared to PrivFT's 91.82% and 89.88%. When it comes to the DBPedia dataset, the BERT IPTP algorithm scores highest at 99.39% and PrivFT scores at 98.47% accuracy. However, PrivFT is an efficient way to apply the FHE scheme and ensure security but it takes a huge time of 5.04 days for training for the tested datasets. Also, it has been designed to receive user-encrypted data and infer classification on the cloud. Because of limitations, it has not been materialized. In the future, with proper improvement, the limitations could be overcome.

Keerthi and Surendiran (2017) [16] have modified the Elliptic Curve Cryptography system by introducing a new mapping technique of encoding messages into affine points on the elliptic curve. Elliptic Curve Cryptography (ECC) is a public key cryptographic system where texts are encrypted using the private key of the sender but decryption is done using both the sender's public key and the receiver's private key. The proposed method of encryption using elliptic curve cryptography is a great contribution to enhancing security and privacy. Elliptic curve cryptography is a popular method in encryption and with a smaller key size, it provides better security. The prime operation in ECC is scalar multiplication compromising point addition and point doubling. The encryption is done by mapping the message into points on the curve and then performing the scalar multiplication using the sender's private key. In this research, the authors have found a new secured mapping technique that maps the messages to affine points on the curve. Point addition and point doubling create the building blocks on the elliptic curve. These two basic operations are done in the Jacobian coordinate system and after point inversion two-dimensional points are converted into three-dimensional projective coordinates. The most vital point in Elliptic Curve Cryptography is the key generation because for encryption it needs both private key and public key and they should know the chosen elliptic curve. For a secured communication channel, the sender encrypts the message using the receiver's public key and the receiver decrypts the message using his/her private key. In this new method, to map the plaintext into affine points on the curve, at first, each character of the plaintext is converted into ASCII values and then ASCII values are converted into HEXADECIMAL values. After that, based on curve parameter grouping is done and goes through scalar multiplication in reverse order. This research is done in a computer having a specification of Intel i3 processor, 2GB RAM and 192-bit NIST prime recommended elliptic curve parameter. The proposed method has been able to improve the performance in execution time while encryption time is 0.08s and 0.03s for decryption time. However, further improvement can be investigated in future works.

Omolara et al. (2019) [22] proposed a modified honey encryption technique for natural language communication. The researchers used a dataset of natural language messages to create and test the improved honey encryption technique. Their suggested approach included extensive testing of the scheme's performance. They investigated aspects such as word count, noise introduced, word count variations throughout decryption, and the existence of significant and typical phrases from the underlying plaintext in the decoded message using controllable noise to the ciphertext and analyzing the sanity of the results. Their technique enabled them to fully evaluate the scheme's efficacy in protecting natural

language communications. The study found that using modified honey encryption techniques improves the security of natural language transmissions.To enhance the revised honey encryption technique, their work may also entail testing and extensive user input. They may also investigate useful uses of the tactic in actual communication networks.

Biswas (2020) [28] examines the privacy issues that are becoming more and more prevalent as chatbots are used in more industries. Although earlier studies have mostly concentrated on improving NLP accuracy, the suggested remedies provide privacy-preserving features. The initial method entails privacy screening and transformation based on "entity" and is suitable for clients who are aware of the chatbot's design. The second technique, which is based on Searchable Encryption, protects user privacy in chats without requiring the user to understand the internal workings of the chatbot. These methods help to protect user data in the changing chatbot interaction landscape across several businesses. In the 'entity' -based privacy protection strategy, alongside 'intents' and 'utterances', giving 'entities' is vital for customizing chatbots. These methods provide a substantial contribution to the protection of user data in the constantly changing chatbot landscape across several sectors. They successfully completed both API requests in the allotted 2-second response time, in terms of performance. The only negative aspect is the higher expense resulting from the additional PPCM API request. However, cost worries can be reduced, guaranteeing a sustainable and economical implementation, thanks to the availability of open-source NLP/Chatbot engines like RASA and the decreasing cost of chatbot API calls. A Searchable Encryption (SE) based technique is developed to ensure user conversation privacy in cases where the chatbot implementation is closed, especially with external chatbots. This strategy does not require knowledge of the chatbot's architecture or NLP engine algorithms. Searchable Symmetric Encryption (SSE) and Public Key Encryption with Keyword Search (PEKS) are the two main branches of SE, which protect sensitive data while allowing server-side searches. This article focuses on PEKS, which limits the generation of trapdoors to the owner of the private key while enabling users with the public key to generate ciphertexts. These solutions are meant to mitigate growing privacy threats and promote greater uptake of chatbots in the company.

Text embeddings exhibit vulnerability in numerous applications, and to prevent external and internal malicious user attacks, the research by Kim et al. (2022) [39] focuses on homomorphic encryption (HE) using the CKKS (Cheon-Kim-Kim-Song) method supporting approximate arithmetic operations. Documents in privacy-sensitive industries such as finance, which are stored in a centralized server are exposed to privacy threats, so service users can encrypt their query which may contain sensitive information and when it is sent to a server it can now perform computations without decrypting the data preventing inversion attacks. With the support of the HE-based text similarity function, the server can transmit the encrypted result after computation to the user where text embedding can be restored to the original message using a secret key by the user. With the help of two text similarity tasks, STS (Semantic textual similarity) which will evaluate semantic similarity, and text retrieval which will evaluate the text retrieval quality of the correct document that is to be searched, the author evaluated their approach. Using the correlation evaluation toolkit SentEval they evaluated the ability to guess the semantic similarity with the STS dataset and computed cosine similarity between text embeddings. The BEIR benchmark was employed to evaluate the performance of text retrieval for a document that needed to be searched based on the user's query. The black-box inversion

method, where the attackers cannot access the model, was chosen to investigate inversion risk in text similarity tasks without the HE approach and so 1-layer MLP was trained as an inversion model inputting text embedding. However, the model's performance was poor, as it was designed to excel only in extreme multi-level classification scenarios. Additionally, the dataset with longer search queries had the worst performance. Using the lowest noise ($\eta = 175$) the model starts to confuse semantically related words; conversely, when the noise is maximized, it fails to extract any words. This experiment shows that text embeddings have information leakage vulnerability. Therefore, to prevent embedding inversion, utilizing an approximation of the square root inverse they implemented the HE method. Here CKKS supports encrypted arithmetic computations. Additionally, because noise limits the computational performance, bootstrapping operations were implemented to increase the number of operations that can be performed on the ciphertext. With their method, the results on the text retrieval dataset show robust performance, also the STS dataset preserved the plaintext performance, so these methods do not harm the model's performance. However, they failed to achieve the HE-based efficient method since encrypted operations can be computationally expensive and complex operations like hashing and graph-based search were not possible as here they use simple cosine similarity functions. Thus, this leads them to one of the future works of implementing efficient search methods of homomorphic encryption.

One-dimensional convolution neural networks (1D-CNN) were used by Wang et al. (2017) [17], to develop an end-to-end encrypted traffic categorization technique. This study aimed to create a unified framework for automatic learning of nonlinear connections between raw input and predicted output by integrating feature extraction, feature selection, and classification. The authors utilized the public ISCX VPN-nonVPN traffic dataset, which comprises 14 types of encrypted traffic, as their test dataset. They investigated four types of traffic representation and utilized the first 784 bytes of each session as raw traffic, and just the first n bytes of each flow to train their model, which was built on the TensorFlow software framework. For VPN-VoIP traffic and VPN-Email traffic, the authors' accuracy rates were 99.5% and 80%, respectively. The proposed approach gets rid of the necessity for conventional phases that are frequently employed in divide-and-conquer strategies, such as feature design, feature extraction, and feature selection. However, the authors propose to investigate the optimal byte count for various traffic classes and tackle the problem of unbalanced training data.

With the support of auto-regressive language models which assist in memorization, Stevens and Su (2023) [43] have proposed a symmetric encryption algorithm named SELM, a probabilistic cipher using generic hybrid construction. According to the research, this is the first study to explore the memorization ability of a language model on random subspaces and arbitrary data. Here, the two participants, Alice and Bob with the help of a secret key encrypt and decrypt messages, whereas the third party Eve does not get access to the secret key. To encrypt a message using the SELM Model, the first step is to convert the tokens where the tokenizer must be invertible, with a random prompt that the model has not seen before, prefixed within them, to reset biases. This helps speed up the memorization process. Next, the model needs to generate a projection P_k where, in the decryption process, the same projection will be using the same key (k). Alice stops the training when the language model generates correct messages given a conditioned prompt. Then the prompts are converted to cipher texts simultaneously the message

encrypted is represented as a d-dimensional vector θ_*^d as an end result here, the number of trainable parameters in the language model is denoted by d. Lastly, Bob with the help of these prompts de-tokenizes the tokens by projecting θ_*^d on the fine-tuned language model which Alice previously used. They utilized the dataset of news articles XSum and the GPT-2_{small} model. Some of the interpretations they have come up with are that language models can fully memorize random data when optimized in a random subspace. Additionally, the model's ciphertext depends on the number of trainable parameters d meaning, the smaller, the d, the smaller the ciphertext where encryption takes a much longer time. To aid in memorizing more complex data, intrinsic dimension parameters can be added. Lastly, the model can memorize messages better when it is pre-trained. SELM can encrypt any message due to its language model's ability to memorize random noise. To ensure security, they have tested their proposed novel symmetric neural cipher SELM, with a modified security game IND-CPA (Indistinguishability under chosen-plaintext attack) by training five models with different message domains and ciphertext distributions and losing the game indicates that Eve is not learning *something* from the ciphertext and cannot decrypt. With the help of regularization which minimizes ciphertext distribution differences, the regularized variant improved SELM's security performance. The authors interpreted as they have used Wasserstian-based regularization which may have slowed the algorithm's speed. However, SELM is not fully semantically secure, and some potential solutions may be:- stronger regularization which will help in improvement in memorization, large language models can memorize better, and longer prompts in tokens boost up the memorization process. Lastly, for future work, improvement of the algorithm's speed can be made by adapting to higher-quality hardware support.

Podschwadt and Takabi (2020) [33] emphasized the importance of security and privacy in (NLP) natural language processing. In their work they tried to classify word embedding using (RNN) recurrent neural network and homomorphic encryption. With this technique, they tried to perform a safe sentiment analysis on the sensitive data. In their research, Podschwadt et al did sentiment classification on a dataset that is related to IMDb Movie Review. Here, with the help of (RNN) recurrent neural network they used a process called CKKS where the homomorphic encryption can be used to process the entail dataset. For this, they trained the model. To train the model they have used ambiguity-free text that helps to predict through encryption without dropping any kind of information. They have used RNN because it can maintain a decent accuracy apart from just encrypting the data. Also, for the activation function particular procedures have been used for low-degree polynomials. The researchers have demonstrated that Podschwadt and Takabi (2020) could make substantial use of RNN models for encrypting information, while it has not been demonstrated how this activity should not compromise the correctness of the procedure by misstating them. Innovative investigation made possible when homomorphic encryptions are combined with RNNs in order to safeguard privacy in natural language processing. Secure machine learning shows a lot of potential. Nevertheless, deriving phrases may turnout as quite time-consuming. This must be corrected through better batch processing optimization. Podschwadt and Takabi (2020) are passionate about researching LSTM as well as GRU structures. Either expand secure machine learning applications or seek superior solutions for dealing with noise in processing encrypted data. Their goal is to help privacy-friendly machine learning in NLP.

Incorporating natural language processing and differential privacy in their work, Panchal et al. (2020) [32]came up with plausible Honey Encryption Scheme decoy communications that reinforced message security. One of the things likely to give malware creators a headache as they spy on text is how the official and fake texts are almost indistinguishable. These principles which border on language analysis and differential privacy lead to false messages whose reality is shrouded in ambiguity, making them seem genuine. The research was carried out to predict the privacy of text documents using machine learning. Researchers used unprocessed instructions that were in form of words, not numbers to train The machine learning native language processing algorithms The text analysis entailed word embedings, transformers models, bag of words, context categorization and keyword extraction. Because language analysis as well as differential privacy schemes are some of the tools commonly employed in constructing fake documents which can sometimes resemble real ones but do not disclose their make up, it makes sense to assume that such methods would work for machine learning research. Panchal et al (2020) have used this methodology to create imitations of genuine communication patterns. The output produced by using this technique is what is referred to as Decoy SMS, as opposed to authentic SMS. Other messages have been specifically created for harmful motives such as deception. Decoy messages always carry a consistent idea and greatly resemble a real one. Panchal et al. (2020) coupled natural language processing with privacy differentiation in an innovative manner. This makes communication between two parties safer and more reliable in encryption. In a similar vein, scientists utilized decoy messages under the Honey Encryption Scheme to safeguard data. Even though it sounds good, the results of the research might be limited due to concentrating emphasis on one side or aspect of communication in constructing an artificial model system that is based on few individuals. Trying out novel ideas may present difficulties particularly with regard to issues whose meanings seem different but remain unclear. On the other hand though; using cutting-edge data privacy algorithms together with those for security has made it possible for expanding a new technique. Because of this, there may be a need for a large number of different sorts and shapes, which will result in a variety of different industries. Not to mention the difficulties associated with dealing with privacy regulations and natural language processing technology, all of which need algorithms that are becoming more intricate and make scaling these computations a genuine difficulty.
Furthermore, it is essential we et page this gap. It could be useful in further studies to consider the use of unsupervised learning methods in extracting context from messages so that specific topics are not solely dependent on massive body corpora. This would reduce the time taken when processing content and make it easier to apply under different scenarios. Furthermore, minimizing author identification probabilities would improve security and privacy of the decoy message generation process. This is more severe when the opponents have enormous sums of publicly accessible information concerning the authors themselves.

In the exploration performed by Kushwaha et al. (2018) [18], on Selective Encryption using Natural Language Processing for Text data in Mobile Ad hoc Networks, is a way of improving security in data communication using wireless. Here, in the study, researchers tried to implement and evaluate the (SSDE) Selective significant Data Encryption algorithm. Kushwaha et al did text data mined from Mobile Ad hoc Networks. It was engaged with the SSDE that can be used for selective encryption. In the process, any irrelevant symbols that were present in the message was removed, and these broke down

into individual tokens. While using Python in this research, the NLTK package was used to process the messages and implement encryption which was usable for some messages to encrypt while ensuring security. Especially, in the wireless networks it was quite useful. Also, the researchers found the SSDE algorithm can improve the data protection system significantly. Encryption time was reduced since only important information was encrypted while common words remained unencrypted. Network performance was improved and reliable data protection was provided due to a certain degree of uncertainty. SSDE outperforms standard encryption for wireless communication, according to thorough testing. Kushwaha et al. (2018) noted that SSDE reduces computational complexity in data encryption and decryption, improving network efficiency. Wireless security and data transfer scale are improved by selective encryption using natural language processing. This technique is only compatible with textual data, therefore it may not work with other sorts of data. Extensive experiments have demonstrated that SSDE is superior to traditional encryption algorithms for wireless communication scenarios.
However, Kushwaha et al. (2018), one key feature of SSDE is its capability to minimise computational complexities during data encryption and decryption operation hence improving network efficiency. Also, employing NLP methods in selective encryption allows for more scalability in terms of how information is sent as well as increases security in wireless communication on the whole. Nonetheless, a drawback that this technique faces is only being able to work with textual information as it currently does; hence, it might not work well with other forms. Future research should broaden the SSDE method to encrypt file formats other than text, say Kushwaha et al. Selective encryption in multiple networks and testing its efficacy may improve wireless data security.

In summary, through our exploration of sentiment analysis (SA) and related topics, we've learned from Wankhade et al. (2022) about many SA methodologies, such as lexicon-based techniques, machine learning, and hybrid methods. Sosea and Caragea (2021) introduced us to Emotion Masked Language Modelling (eMLM) which is a BERT-based approach that targets biases unique to sentiment. Saju et al. (2020) offered a comprehensive analysis of sentiment analysis that took into account ethical issues, tools, and APIs. Acheampong et al. (2021) discussed about the challenges of figurative language when it comes to detecting sarcasm. To examine the user tone in order to identify depression, Tadesse and colleagues (2019) have contributed in the healthcare domain by involving linguistic dimensions, psychological processes, and personal concerns (LIWC) these three features. COVID-19 related tweets have helped Imran et al. (2020) to examine emotion and cross-cultural interaction by using three step model training process with the help of word embeddings and neural networks. Acheampong et al. (2020) emphasized on the use of emotion detection which are text based and the possible application of it in analyzing depression. Utilizing semantics-based feature vector, ensemble classifier method to improve sentiment analysis result was proposed by Kanakaraj and Guddeti (2015). Using two datasets and two classifiers (SMO and J48) drawbacks of emotion detection and an improved method were discussed by Gaind et al. (2019). Based on tweets Chong et al. (2014) explored sentiment analysis on social media. Improvement in sentiment analysis can be made by combing domain knowledge on language models was presented by Ying et al. (2019). In encryption method, we have learned about Feng et al.'s (2020) SecureNLP which demonstrate PrivLSTM enhance privacy-preserving seq2seq models. PrivFT, an effective completely homomorphic encryption technique for text classification, is presented by Badawi et al. (2020). It has competitive accuracy but requires a

lengthy training period. Keerthi and Surendiran (2017) present a secure mapping method that improves elliptic curve cryptography. Honey encryption system by Omolara and Jantan (2019) can be effective in safeguarding messages in the field of NLP. Biswas (2020) presented that "entity"- based encryption can help in tackling privacy when interacting through chatbot tackles chatbot privacy. Kim and colleagues (2022) discovered that text decryption process is not needed in the other end when homomorphic encryption based on CKKS is used. An end-to-end encrypted technique proposed by Wang et al. (2017). Finally, Stevens and Su (2023) proposed SELM, a symmetric encryption technique with the help of language model memorization.

Chapter 3

Work Plan

Our research aims to build and evaluate a secure communication system employing substitution encryption, and then create and test a machine-learning model for text interpretation. At first, the text "hello there" undergoes substitution encryption on the client side at the beginning of this process. In our technique of substitution encryption, certain letter pairs are replaced according to a predetermined dictionary. For example:

$$h \rightarrow sc, \quad e \rightarrow EZ, \quad l \rightarrow > u, \quad o \rightarrow <>$$

Prior to send from client side, the encrypted text will undergo tokenization, that is the tokenization of the encrypted text into smaller units such as words or subwording for enabling better processing or analysis through the deployment of machine learning algorithms. Then, to further preprocess the model after the tokenization, the sequence and padding are done to make them all the same length for equal input size.

In the embedding layer, using clean data from the corpus named Wiki512, the embedding layer takes the tokenized text and converts the words into numerical word embeddings. The data is cleaned by only using lowercase letters (a-z) and digits (0-9). Utilizing these embeddings, it maintains semantic relations of the words and their interconnections. It grasps the connotation of each word in relation to the other words in the concept. It also allows for a more contextual view towards any encryption and tokenization of text. Nevertheless, it improves the information in order to prepare it for the training of the models.

The model training layer comes after the embedding layer. The embedded and tokenized data are used to train an LSTM or GRU model in this procedure. These models are specifically tailored to recognize patterns inside sequences. They work especially well with encrypted text material that has undergone this kind of processing. The model's objective is to reliably and securely interpret confidential talks.

At last, training of the performance evaluation of the model includes the accuracy and the F1 metrics scores. Accuracy defines the ratio of the number of correct cases within the total amount of cases but the F1 score is expository for the effectiveness of the model since it takes into account both precision and recall. Such massive techniques application adopts a workflow whereby the communication system achieves security, efficiency and text understanding in spite of various stages of hashing, encryption and embedding, and training of the model. In this manner, we address secure encryption; complex machine

learning; and performance assessment to arrive at communication systems that enable the users' privacy but also thorough text comprehension.

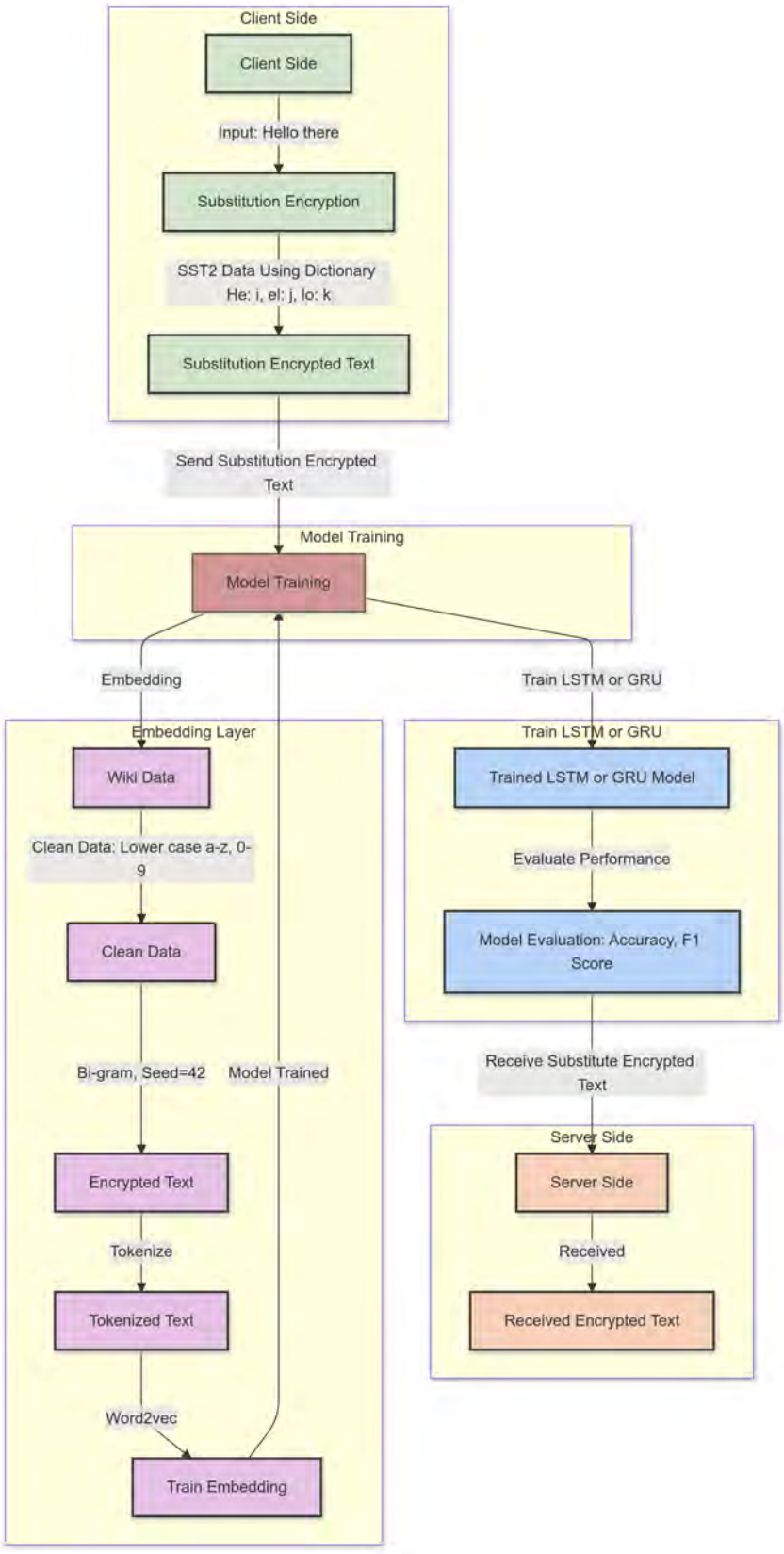

Figure 3.1: Work Plan

Chapter 4

Methodology

4.1 SST2 Dataset

The Standford Sentiment Treebank 2 (SST2) [8] has been selected as the primary dataset to train the models. The Stanford Sentiment Treebank is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language.

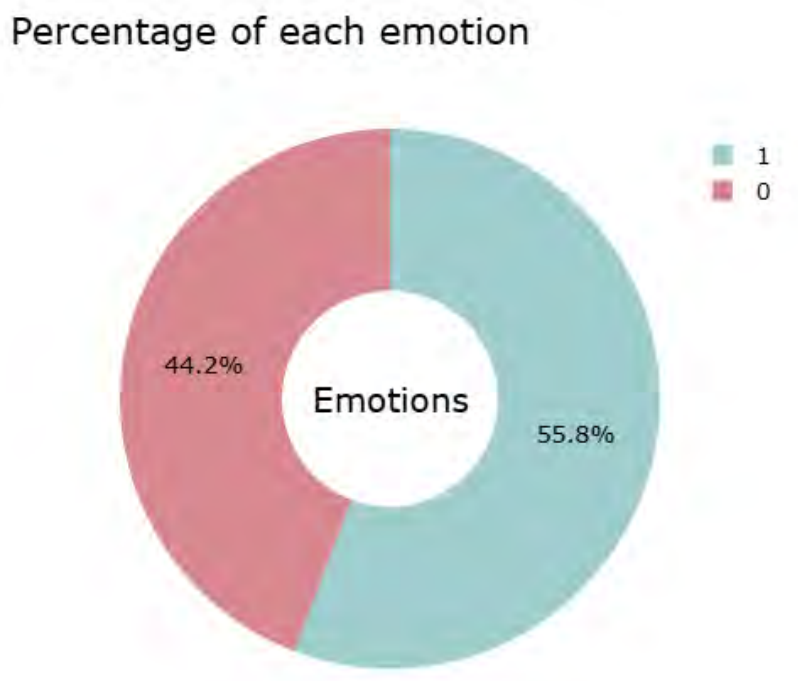

Figure 4.1: Percentage of Each Emotion

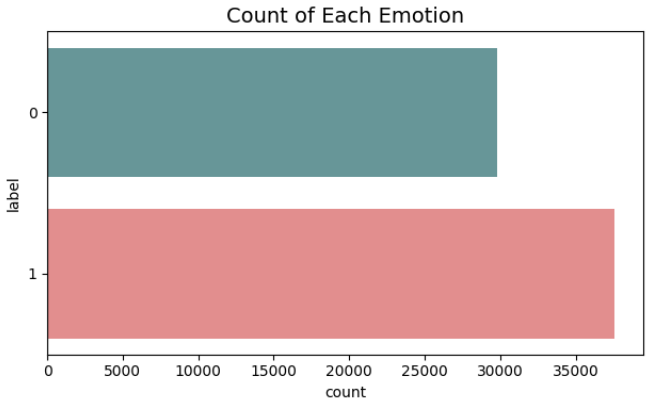

Figure 4.2: Count of Each Emotion

This dataset contains two types of emotions which are negative and positive. This dataset contains 67349 sentences which are mainly collected from movie reviews and each has been annotated as negative or positive by three independent human judges. 56% of the data with 37569 sentences labeled as positive emotion while the remaining 29780 sentences, in percentage 44% data having negative emotion.

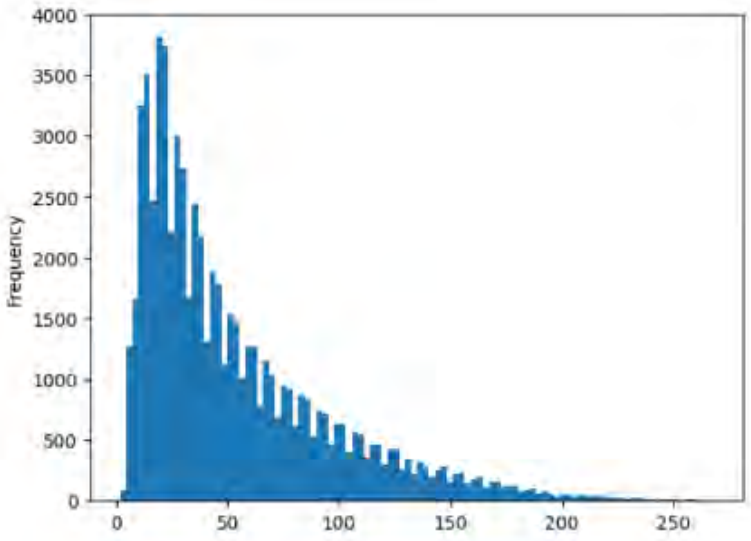

Figure 4.3: Sentence Length

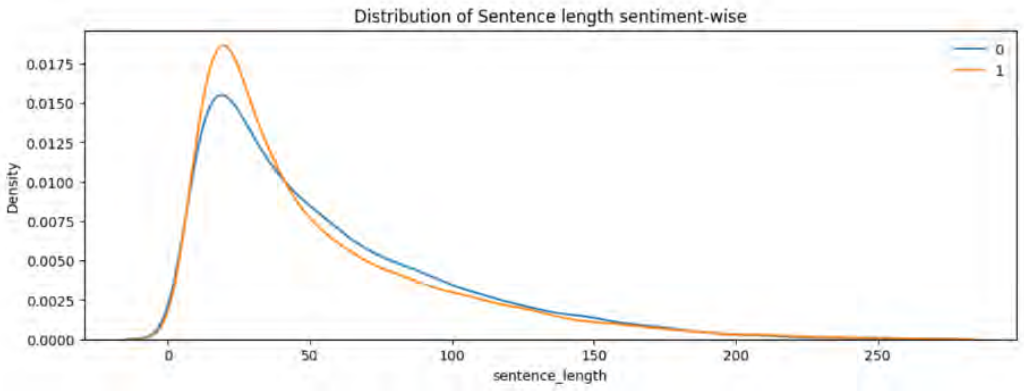

Figure 4.4: Distribution of Sentence Length Sentiment-wise

The distribution of sentence lengths shows a clear trend that shorter sentences are significantly more common than longer ones. The highest frequency is observed with sentence lengths close to zero, indicating that these shorter sentences occur most frequently. As the sentence length increases, the frequency decreases, indicating that longer sentences are progressively less common. The majority of data points cluster around shorter sentence lengths with fewer instances of longer sentences.

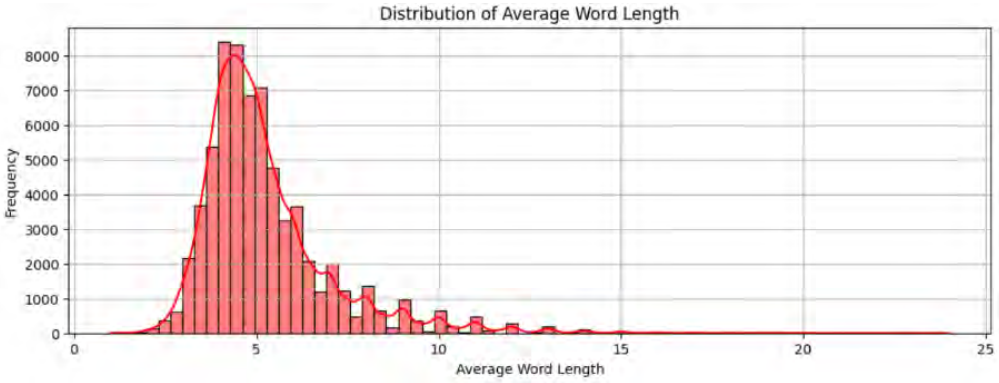

Figure 4.5: Distribution of Average Word Length

The average word length of a sentence is mostly distributed between 4 to 5 with a frequency of 5500+ to 8000+ rows. Even sentiment-wise words are distributed in almost the same order.

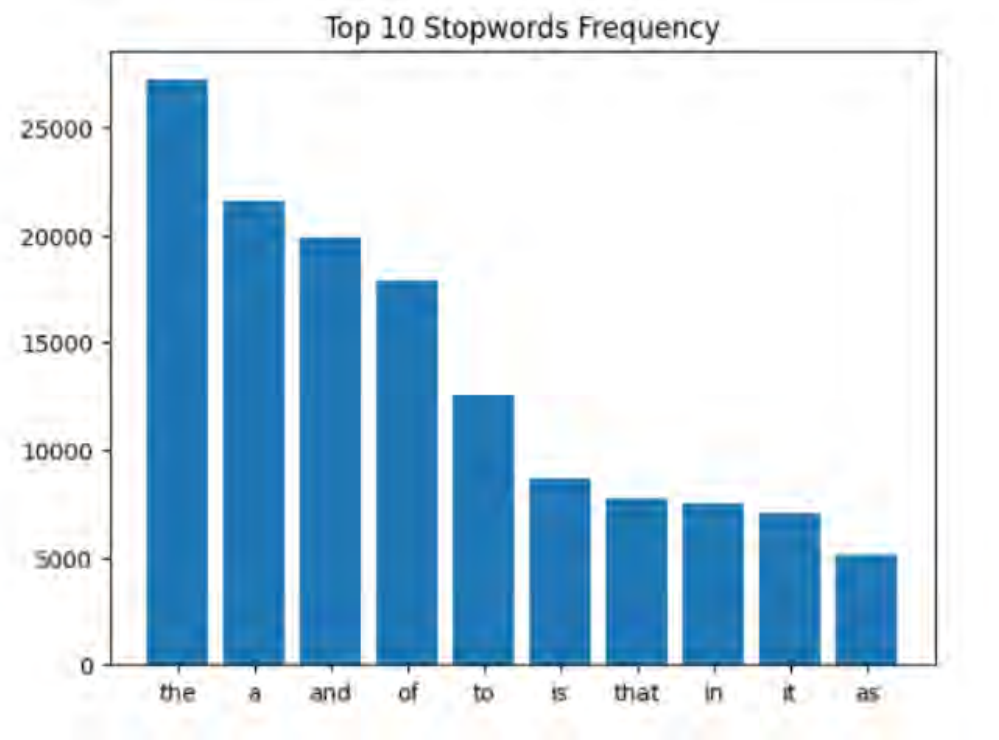

Figure 4.6: Distribution of Stopwords

The most frequent stopword is “the” with a frequency exceeding 25,000 occurrences. This is followed by “a” and “and” both of which have frequencies slightly above 20,000. The word “of” appears nearly 18,000 times, while “to” has a frequency of just over 12,000. Other stopwords in the top 10 include “is”, “that”, “in”, “it”, and “as”. These words have

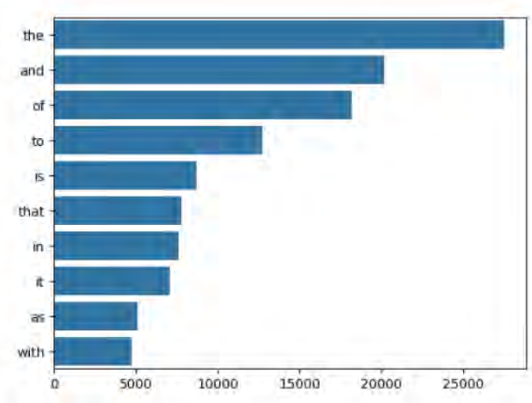

Figure 4.7: Unigram

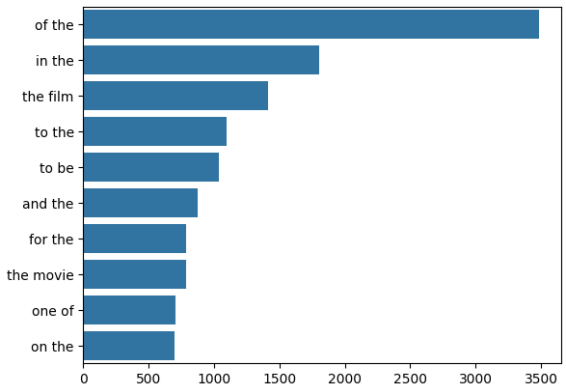

Figure 4.8: Bigrams

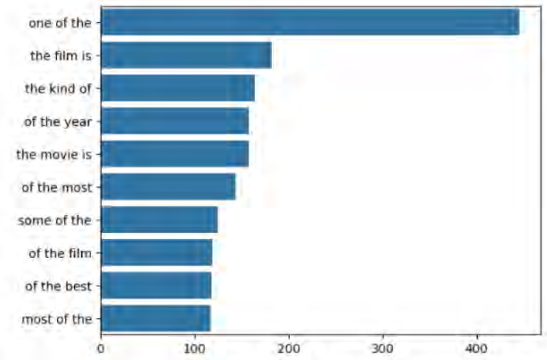

Figure 4.9: Trigrams

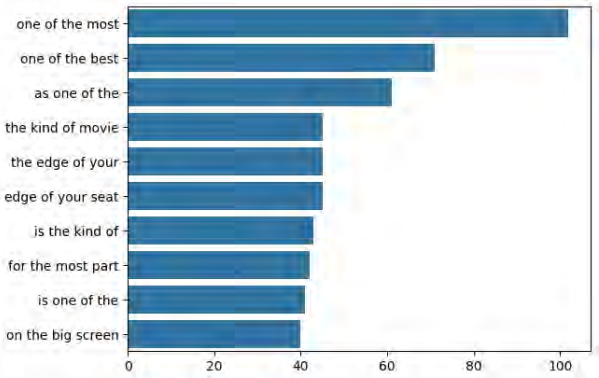

Figure 4.10: Four grams

frequencies ranging from around 9,000 to 5,000. Specifically, “is” appears approximately 10,000 times, “that” and “in” each has around 8,000 occurrences, “it” has about 7,500, and “as” appears close to 5,000 times.

Analyzing n-gram visualization, showing the frequency of different phrases in sentences. The longest bar corresponds to the highest frequency of that phrase. The diagram provides insights into common linguistic patterns used in the containing sentences of the dataset.

To note, the official dataset SST2 has been collected from the Hugging Face using the Hugging Face API.

4.2 Wiki Dataset

The Wiki dataset [49] under analysis seems to have ample amount of text data with more than 6.6 million records, where each record is in the form of a row in a DataFrame. The data is in fact placed within single column which is titled as ‘text’ and every entry in this column is with different word count. This indicates that the text entries are likely to be mostly of smaller sizes which is further supported by the word length count distribution seen in the scatter plot attached.

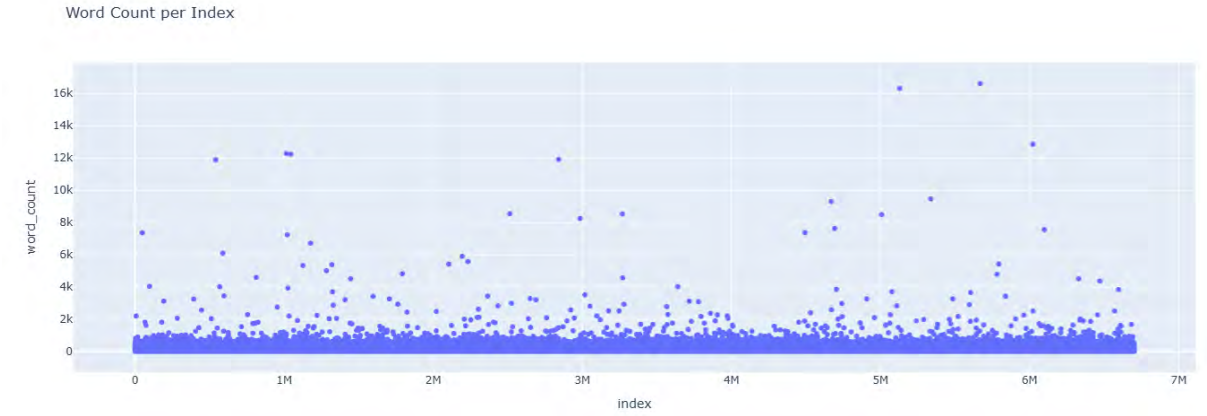

Figure 4.11: Word Count vs Index

Word count per index for the data subset is represented as a scatter plot in figure 4.11. The majority of text entries tend to be quite short, huddling together between the 0 and 2,000-word mark, which implies that short pieces like a few sentences or paragraphs are most common in the dataset. Certain notable outliers with word counts going beyond 16,000 also suggest that longer documents were included as well. The greatest word count of 103,794 words emphasizes the degree of diversity in the dataset coupled with sharp peaks that compass entries with extremely long, or short, words than the rest of the dataset.

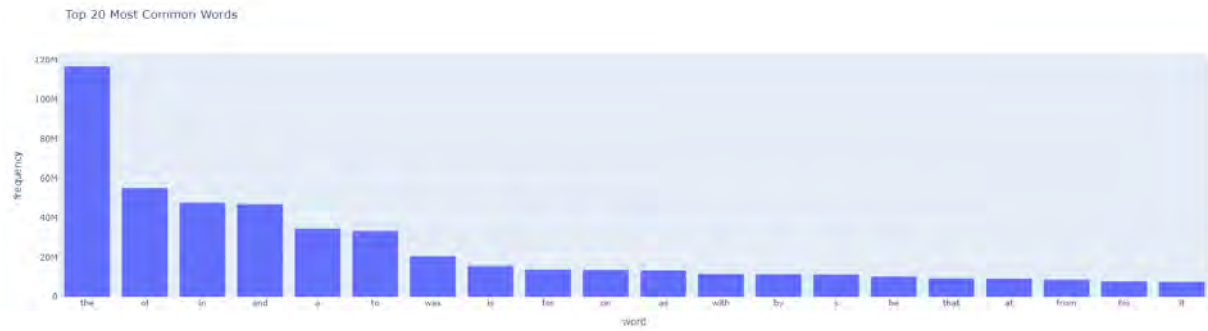

Figure 4.12: Top 20 common words

The dataset comprises more than 6.6 million wikipedia entries/articles/paragraphs, which have been recorded into the system in one column, and the maximum character length in a sentence is 10,000+ while the minimum is 6. Such a wide span indicates that various types of texts have been included in the corpus, from single line reviews to articles or even books. Average tokenized length is 512 for each sample. Maximum word count in a sentence is 16000+ while the minimum is 1. High rates of stopword frequency illustrate that the information is probably written in a conversational language thus pointing out the use of some pre-processing strategies like tokenization or stop-word annihilation, to obtain meanings and highlight the issues in the dataset. Highest encountered word is 'the' which is close to 120 million times. To note, the official Wiki dataset [49] has been collected from the Hugging Face using the Hugging Face API.

4.3 Encryption Methods

4.3.1 Substitution Cipher

Using a substitution cipher, the ciphertext is created by substituting different pair of letters for each one in the plaintext. Every letter in the alphabet can be replaced with a pair of other letters in its most basic form. When the 'h' in the sentence "hello there" is replaced with 'sc', for example, all 'h's in the plaintext will become 'sc' in the ciphertext. In several applications, including ASCII code, shorthand, semaphore, and Morse code, substitution ciphers are utilised. When frequent letters and sequences in the ciphertext are compared to established linguistic patterns, substitution ciphers are susceptible to statistical attacks.
A substitution table or key that associates each letter in the plaintext with its corresponding letters in the ciphertext is constructed in order to encrypt using a substitution cipher. The matching letters from this table is then used to replace each letter in the plaintext. Decryption uses the same substitution table to reverse the operation.

The Substitution Dictionary that has been generated for this research is as follows:

Original	**Substitution**	**Original**	**Substitution**	**Original**	**Substitution**	**Original**	**Substitution**
a	r}	b	%ˆ	c	W˜	d	n_
e	EZ	f	˜E	g	vT	h	sc
i	W&	j	68	k	{̃}	l	¿u
m	¿R	n	A”	o	¡¿	p	e*
q	Fk	r	’X	s	0L	t	]O
u	w’	v	jU	w	tK	x	kf
y	[f	z	hD	A	Fn	B	j{
C	Uz	D	s—	E	_*	F	_.
G	o&	H	—?	I	IS	J	8T
K	-x	L	w˜	M	gk	N	—Y
O	dV	P	E(	Q	v{	R	&R
S	2)	T	Pg	U	‘X	V	1H
W	iH	X	dT	Y	Nˆ	Z	)n

Table 4.1: Substitution Dictionary for Single Characters

Original	Substitution	Original	Substitution	Original	Substitution	Original	Substitution
!0	=T	!1	VX	!2	'd	!3	KD
!4	b$	!5	:'	!6	/J	!7	ov
!8	wA	!9	D&	!a	b_	!b	V_
1p	VR	!d	W̃	!e	mj	!f	F7
!g	lP	!h	hH	!i	X8	!j	R=
!k	:S	!l	!i	!m	I)	!n	I5
!o	va	!p	HM	!q	Et	!r	¡u
!s	x]	!t	6!	!u	Ho	!v	w7
!w	2r	!x	6s	!y	RQ	1t	Yp
$l	Pw	$m	67	$n	8Z	$o	pV
$p	'\	$q	kr	$r	l*	$s	@W
$t	x+	Sq	Di	Sr	¿s	Ss	a]
St	.L	Su	Vo	1m	Vp	1n	f0

Table 4.2: Substitution Dictionary for Pair Characters

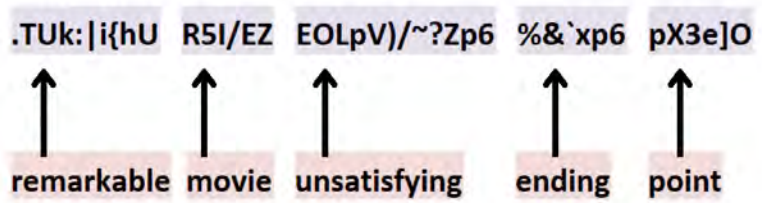

Figure 4.13: Plain Text to Substitution Encryption

Both GRUs (Gated Recurrent Units) and LSTMs (Long Short-Term Memory) are sophisticated RNNs (Recurrent Neural Networks) intended to manage sequential data and address the vanishing gradient issue; LSTMs provide greater flexibility, while GRUs are easier to use and faster. Substitution ciphers are a fundamental type of encryption that is susceptible to frequency analysis attacks since they substitute ciphertext according to a fixed system for portions of the plaintext.

4.4 Embedding Layer

The proposed pipeline's embedding layer is essential for converting textual data into numerical representations that can be processed by machine learning models. This layer is in charge of taking words or tokens and turning them into dense vector representations that capture their contextual and semantic meanings. An overview of the embedding layer is given below:

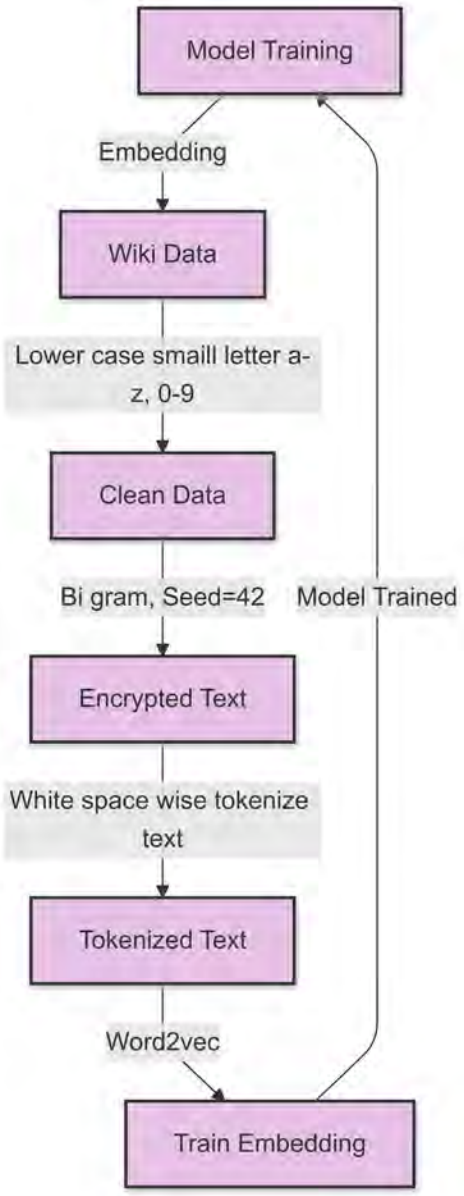

Figure 4.14: Embedding Layer

4.4.1 Text Generation

The preliminary data collection is made up of 6,699,666 entities drawn from Wikipedia. The first stage of this process will be to purify this data to the necessary standard. This involves the deletion of duplicates which means the collection is now comprised of 6,685,332 entities only. In this case, every single sentence is split based on full stops, line breakers, and tabs and then replaced with whitespace. This creates an organized dataset such that every sentence is in its own distinct data point. Eventually, this comes out to 73,763,903 refined sentences for further analysis and processing.

4.4.2 Data Cleaning

Once the first stage i.e text generation, has been finished, the second stage initiates the development of cleaning. Each of the sentences has to undergo cleaning. In this case,

each entry will also undergo text editing to remove any statuesque text and standardize the syntax. In that, the case of the text will be changed to lower case, and other elements like codes, punctuation marks, or spaces will also be removed including numbers and letters a-z, 0-9. The procedure of cleaning results in uniformity and appropriateness of the data for the next stage of processing.

4.4.3 Text Encryption

When the text is broken down into discrete sentences, it undergoes a mixed substitution cipher. A fixed seed with the value 42 is utilized. It guarantees that if the randomization is done numerous times, the same substitution dictionary will be generated every time. This prevents ambiguity and is important in safeguarding the encrypted text which will be examined in different analyses. The process of encryption will comprise looping through the cleaned sentences and applying orders for substituting the plain text for the encoded text where individual characters and pairs of characters will both be taken into account.

4.4.4 Tokenization

Once the data is encrypted the next step is to perform tokenization on the text. Tokenization is done based on whitespace. This means that all of the sentences that have been encrypted will be further subdivided into smaller parts called tokens thus forming an organized structure that will ease further analysis and training processes. Tokenization will ensure each word/symbol is treated as an independent entity which will allow proper input into the embedding model.

4.4.5 Word2Vec

Word2vec is a widely used approach for producing word embeddings. This model encompasses both syntactic and semantic similarities among the terms. A prominent example of vector algebra applied to the learned word2vec vectors is:

Vector(“King”)-Vector(“Man”)= Vector(“Queen”)-Vector(“Woman”)

In this case, the tokenized, encrypted text taken from the wiki512 corpus is used for Word2Vec. The tokenized and encrypted sentences are the training data, however, the words in those sentences are still regarded as distinct tokens for ease of understanding the context relationships. Here, Word2Vec would enable us to model the textual word relationships within the context of certain words that are close to the range of the encrypted words. Word2Vec operates in two modes:

- **Continuous Bag of Words (CBOW):** This approach assumes a target word given its surrounding context. It seeks to optimize the embeddings so that words occurring in comparable contexts have similar vector representations.

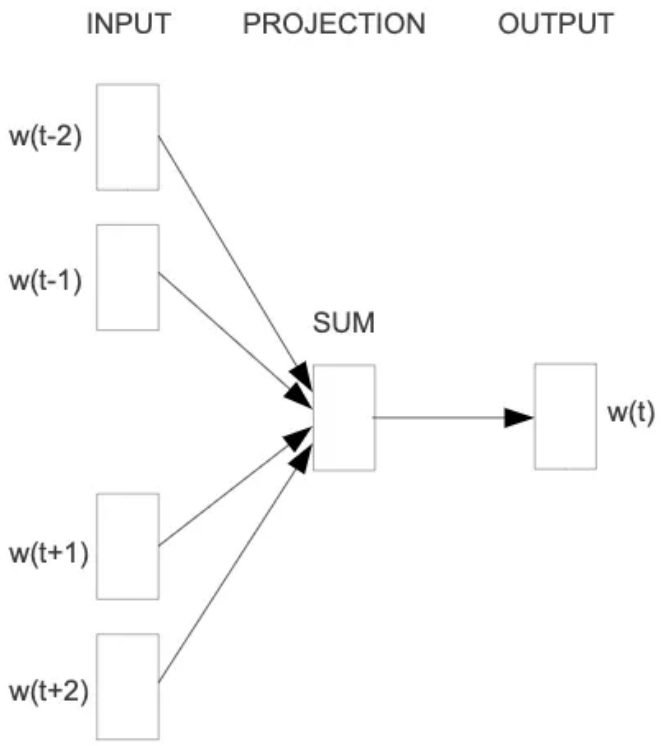

Figure 4.15: CBOW [7]

- **Skip-Gram:** This model reverses the aim of the CBOW model. Given the current word, it guesses the surrounding context words both in the past and future. As the name says, the model predicts the N-gram words except for the current word as is the input to the model, thus the term skip-gram.

$$\mathcal{L}_{\text{skip-gram}} = \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j} | w_t) \tag{4.1}$$

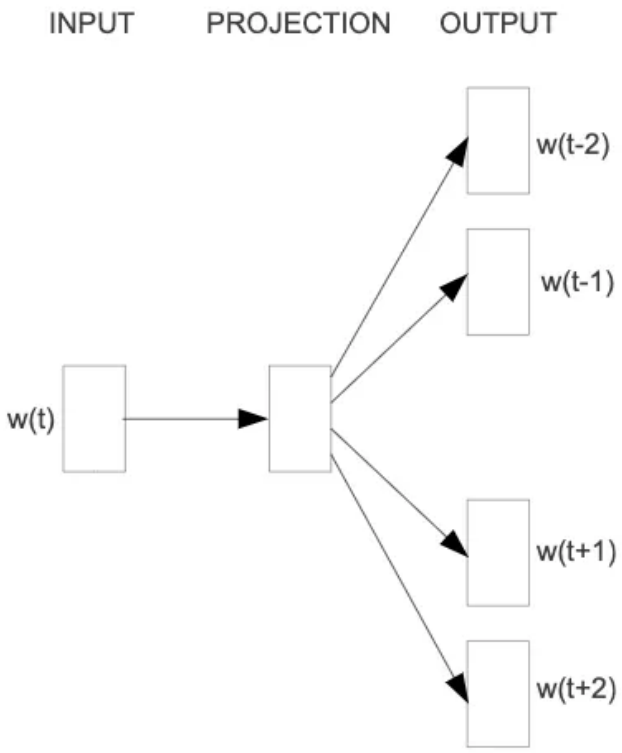

Figure 4.16: Skip-Gram[7]

sg=1 has been set in the Word2Vec model because the primary objective was to develop the model utilizing skip-gram. This is an advantage particularly where the dataset is large with a highly sparse target vocabulary as skip-gram deals with the context of the target words much better than the target words themselves. In this regard, given the size of the dataset of over 73 million clean sentences, the skip-gram model in this regard derives its usefulness because of its capability to predict the surrounding words of the target word. Hence, it is worth highlighting the significance of such predictions considering the presence of a lot of low-frequency words or tokens.

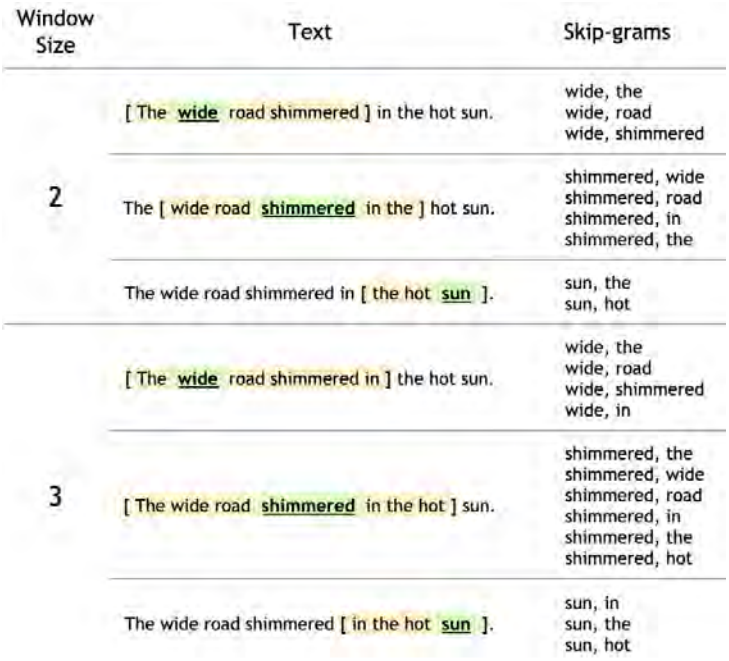

Figure 4.17: Window Size [46]

For parameters, the vector size is 256, which is the word vector size. A window size of 5 was employed in order to capture the context better. According to [46], the size 5 (median = 0.8956) is the best suitable window size. In addition minimum frequency is 35 and the number of CPU threads used in the training was 24. Maximum context length means the number of words on both sides of the target word which can be considered context words or words. This is how training of the Word2Vec model results in a set of word vectors that are sensitive to the encrypted words and their contextual relationships.

4.5 Models

Recurrent Neural Networks (RNNs) are a type of neural network that is used for processing sequential data, such as text, audio, or time series data. They are able to utilise dependencies and context across time steps as a result. RNNs come in several forms, such as Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) neural networks. The issue of "vanishing gradients" in RNNs, which arises when the network's weight gradients are extremely small and the network struggles to learn, is one that LSTMs and GRUs are both intended to solve. The work has been assessed using both GRU and LSTM.

4.5.1 LSTM

Errors such as exploding or disappearing gradients during backpropagation, which cause the weights to become either too large or too little, hinder the ability of simple RNNs to learn long-term dependencies. This is addressed by using RNNs of the Long Short-Term Memory (LSTM) kind. For sequential data tasks like sentiment analysis, language modeling, speech recognition, and video analysis, long-term relationships between data time steps are recognised by LSTMs, which makes them perfect.

Compared to basic RNNs, LSTMs are more intricately constructed, featuring three gates: input, output, and forget. By regulating the information flow, these gates enable the unit to choose what data to retain, discard, or transmit. With the aid of this structure, LSTMs are able to learn long-term dependencies and circumvent the vanishing gradient issue, which frequently arises in RNNs and causes gradients to get too small for efficient weight updates. Furthermore, LSTMs are capable of processing bidirectional inputs and variable-length sequences, which are essential for natural language processing (NLP) applications like sentiment analysis, machine translation, and text production. Compared to straightforward RNNs, LSTMs are more appropriate for analysing sequential data because of their reduced sensitivity to the time interval.
Depending on the inputs and the prior cell state, the gates in an LSTM network selectively store or forget information using sigmoid activation functions, which produce values between 0 and 1.
In contrast, LSTMs can learn more complex and long-range patterns and have mechanisms to selectively remember or forget relevant information, making them less prone to overfitting. For relatively simple and short data, RNNs might be preferred, while for complex, long, or noisy data, LSTMs are more suitable, which is why working with LSTM has been chosen.

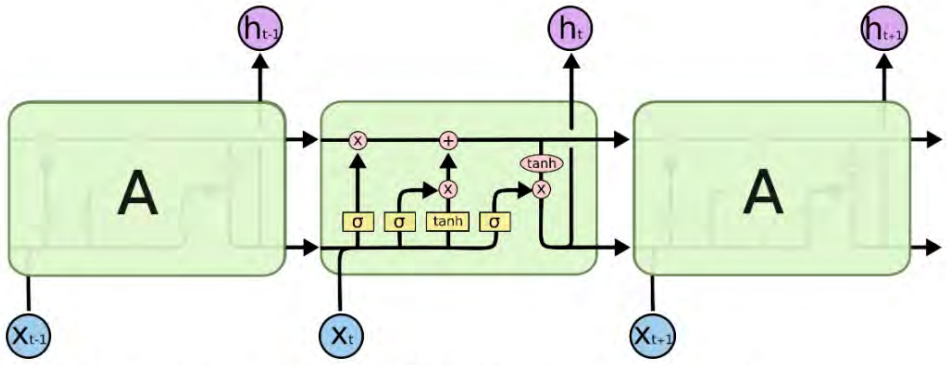

Figure 4.18: LSTM [38]

4.5.2 GRU

Similar to Long Short-Term Memory (LSTM), GRU (Gated Recurrent Unit) is a sort of recurrent neural network (RNN) architecture intended to simulate sequential input by gradually forgetting or selectively recalling information. Compared to LSTMs, GRUs have a more straightforward architecture and fewer parameters, which facilitates training and increases computing efficiency. One element at a time, GRUs process sequential data, changing their hidden state in response to new input and taking into account the hidden state that was previously set. In order to update the hidden state for the subsequent time step, GRUs generate a "candidate activation vector" at each time step by fusing data from the input and the prior hidden state.
The GRU architecture consists of an input layer for sequential data, a hidden layer for recurrent computation, an update gate to govern the amount of the candidate activation vector to incorporate into the new hidden state, and a reset gate to decide how much of the previous hidden state to forget. Using a tanh activation function, the candidate

activation vector is a modified form of the prior hidden state paired with the current input. Depending on the job, the output layer generates the final output of the network based on the hidden state, which may be a single number, a series of numbers, or a probability distribution over classes.
Compared to LSTMs, which have three gates (input, output, and forget), GRUs, which are a reduced version of LSTMs, employ two gates—reset and update. Because of this, GRUs learn and run more quickly. For tasks like speech recognition, language translation, and time series forecasting, both LSTMs and GRUs are utilised. When it comes to activities that call for long-term dependency storage, LSTMs tend to do better than GRUs, which are better suited for jobs that need rapid learning and adaptation. Thus, the GRU model has also been employed in this research.

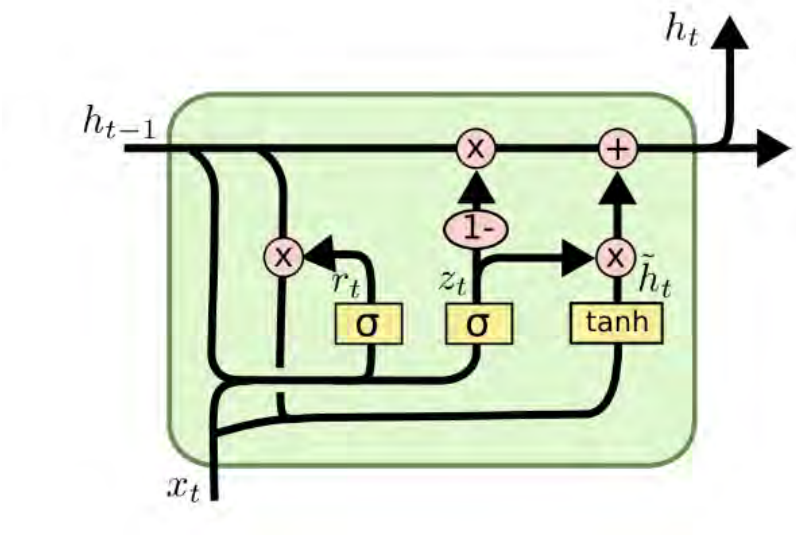

Figure 4.19: GRU [20]

Chapter 5

Proposed Protocol

5.1 The Proposed Protocol

This research is a study on integrating the NLP pipeline with the Encryption module to analyze if emotions are lost while ensuring security. This cross-section of security and NLP examines the significance of encryption in NLP, on conveying emotions in situations where an individual might be in danger and needs to communicate with authorities such as the police via a text-based emergency messaging platform. The transmission from the sender to the receiver is being done using broadcasting and so initially, there is no security. It is crucial to protect this communication's confidentiality and integrity. To protect the initial transmission of the message from the client to the server, a bigram substitution cipher is employed. This method scrambles the message using a substitution table, making it difficult for third parties to decipher the content in the event that it is intercepted. Hence, encryption is being used to protect the user input, by maintaining confidentiality and integrity as no middleman attack is possible as a combination of 8930! cannot be broken by brute force even if possible it will be tough for modern-day supercomputers.

This research also observes the data persistence and preservation of sentiments even after going through bigram substitution encryption with the help of embedding, tokenization, and sequencing. In conclusion, the proposed protocol employs advanced sentiment analysis with LSTM and GRU models to determine the sender's emotional state after substitution encryption is used to safeguard the transmission of sensitive data. This all-encompassing strategy guarantees the confidentiality and authenticity of the correspondence while permitting prompt action by the relevant authorities when needed.

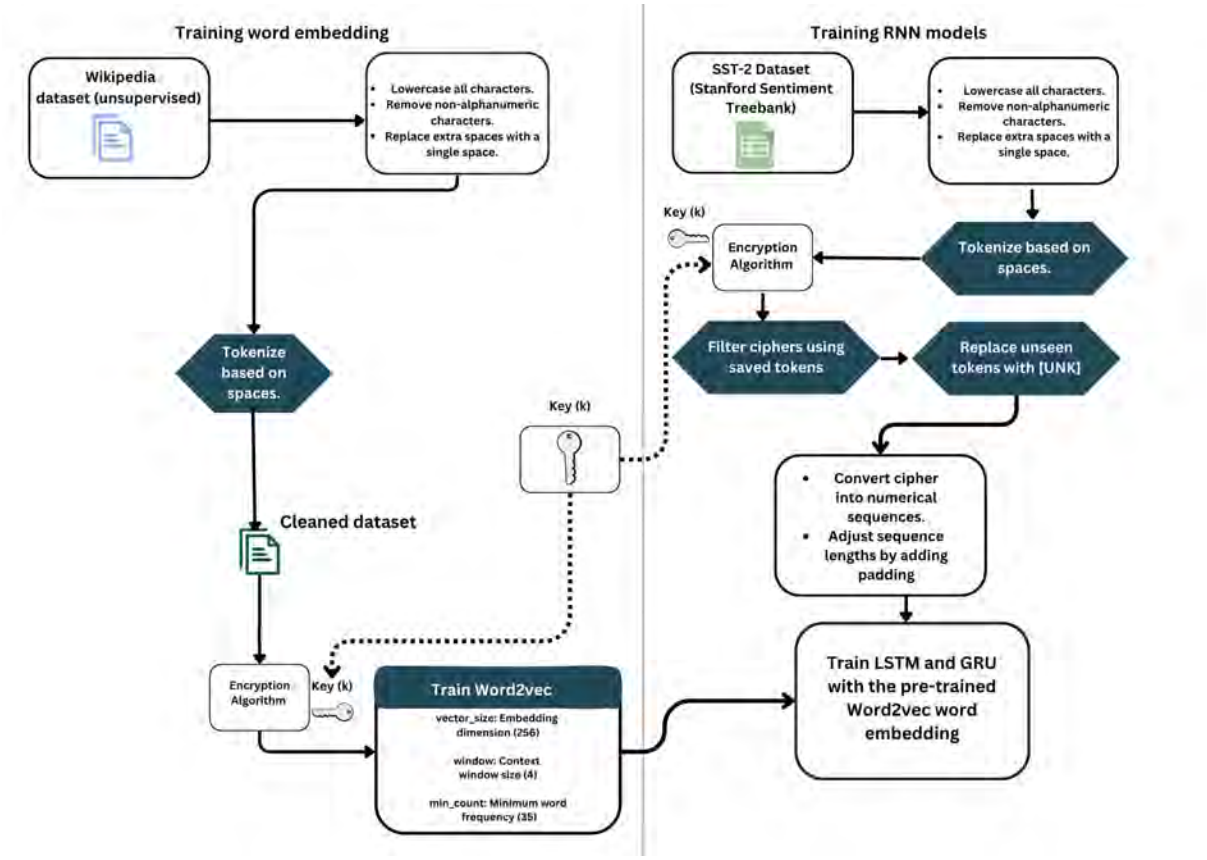

Figure 5.1: Proposed Protocol

This pipeline incorporates an array of components. First off, a polygraphic substitution cipher is used which is a type of substitution cipher where groups of letters are replaced by other groups of letters. Unlike simple substitution ciphers that map one letter to another, polygraphic ciphers operate on multiple letters at a time, making them more resistant to frequency analysis and thus more secure. Due to its ability to regulate the creation of the substitute dictionary, the seed (seed=42) used for randomness is fundamental to the encryption scheme. In fact, the seed practically becomes the key. In order to create the appropriate substitution dictionary and decrypt the message, someone with access to the encrypted message (ciphertext) would also require the same seed value (the key). Because the seed determines how the substitution dictionary is built, it effectively becomes the key to the encryption system. An attacker can decrypt a message by generating the same substitution dictionary that was used for encryption if they know the seed or can guess it. Using seed=42 for randomness, this substitution dictionary is created for both single characters and character pairs. Every time the code is executed with the same seed, the substitution dictionary may be constructed consistently thanks to the usage of a random seed. To make sure that no character or pair is mapped to itself, a substitution dictionary is used.

Each sentence in the DataFrame is first cleaned by removing punctuations, converting it to lowercase, and removing unwanted characters, leaving the DataFrame with only lowercase letters and digits. The original and cleaned sentences are then fed into the sub_encrypt function. Tokenization is done based on whitespace. Both tokenization and normalizing the text before encryption is helpful. The original and modified encrypted

results are kept in a new column called substituted_text.

The character set for substitution includes punctuation(e.g., !@$%^*()), digits(0-9), uppercase (A-Z), and lowercase letters (a-z). The code generates every possible pair of characters, going beyond simple substitution of one character and generating all possible two-character combinations. The square of the total number of unique characters—94 in this case—will determine the overall number of pairs that are formed. As a result, it generates 8930! combinations, as 94(single characters)+8836(bigrams)=8930 are all conceivable two-character combinations of characters. Thus, polygraphic ciphers disperse the frequency over a greater number of combinations, making patterns more difficult to discern. A particular bigram or trigram occurs far less frequently than individual letters, therefore deviations from the predicted frequency make cryptanalysis more difficult.

The cipher creates a substitution dictionary that includes mappings for both single characters (e.g., 'A' → '$s', 'a' → 'dk') and two-character combinations (e.g., 'AB' → '@#', 'Aa' → '$D'). The substitution values are shuffled versions of the characters and pairs. The code ensures that no character or character pair is mapped to itself.The substitution dictionary must stay static during the training and evaluation stages after a proper mapping has been accomplished. This enables the model to correlate specific patterns in the substituted text with the appropriate original meanings during training. This consistency is essential for the model to learn the mappings efficiently. Another factor influencing the reproducibility of the shuffling process is the usage of a random seed. It is guaranteed that the random shuffle yields the same sequence of substitutions each time the operation is executed by setting a specified seed value. On the other hand, the seed can be altered to produce a different shuffling sequence if the device starts to become vulnerable to cyberattacks. By adjusting the seed, one can investigate the model's behavior under various mappings and, in the end, evaluate how well-suited it is to various substitution patterns. Thus, while the seed provides a way to control reproducibility, changing it can introduce variability and test the model's generalization capability.

The cipher initially looks for a match between two consecutive characters in the text and the substitution dictionary during encrypting. In the event that a legitimate pair is discovered, it is substituted, and the pair is skipped by increasing index i by 2. The cipher will continue to function by replacing individual characters with an encrypted pair even if no valid pair is discovered during the encryption process, ensuring that the length of encrypted words is always even. The algorithm's ability to handle inputs of any length is guaranteed by this fallback. During encryption, pairs of characters are prioritized above single characters, allowing increased complexity and security.

Decryption works in tandem with encryption. Using reverse mapping, it first looks for pairings in the encrypted text and replaces them. When a legitimate pair cannot be found, single-character decryption is used by default.

This cipher combines pair-based and single-character substitution: The substitution becomes far more difficult than the traditional single-character substitution when pairings are introduced. This adds a layer of obfuscation and multiplies the possibilities for the encrypted output. The cipher can accommodate a wide range of inputs because it includes both uppercase and lowercase characters, numbers, and punctuation. Traditional

substitution ciphers can be broken by frequency analysis since only one consistent substitution is used for each character. This type of attack is made more difficult by pair-based substitution since the frequency of pairs is less predictable.

The text goes through additional processing in client side after it has been encrypted using the substitution dictionary to send to the server/reciver side. The text is encrypted and tokenized. Tokenization is the act of splitting up the text into smaller units, typically words or subwords, and converting them into a numerical format known as tokens, which are then processed by machine learning models. Here, the embedding layer tokens created based on whitespace—which will be covered later—are used to do the tokenization. Subsequently, a dictionary word index is created so that every token has a distinct numerical ID. Numerical IDs have been assigned after stopwords have been removed. Text was transformed into sequences by transforming it into a series of numeric IDs based on the word index. It should be noted that words missing from the word index are substituted with a UNK token since they are unknown, and then an integer 0 is used to replace them in the sequencing. For the encrypted tokens, a word index dictionary with 0 assigned to unknown terms has been created. Later padding is applied to the tokenized words to a fixed uniform length in order to guarantee that each sequence has the same length for the purpose of training the model. Sequencing and padding are completed at last.
Second, word embeddings are also utilized because they provide a means of encoding words in a continuous vector space by mapping nearby points to semantically similar words. This is done in order to interpret words according to the context in which they appear in the text. The Gensim library's Word2Vec model is used in this code to generate embedding vectors that have 256 dimensions for each word.

For embedding training, the corpus wikipedia_512_pretraining is utilized. For the purpose of pretraining models, t processes the Wikipedia articles.These Wikipedia paragraphs or articles are then split into sentences/sub-sentences which are combined into a new data frame. It is then further cleaned to get rid of duplicates. Prior to training, all texts must be cleaned up by removing special characters, stopwords, and all punctuation and digits to lowercase. After the data is cleansed, a bigram substitution cipher is applied. Here, the embedding is being evaluated via intrinsic evaluation.

Tokenization of the text happens after encryption of the cleansed sentences. Tokenization is the process of breaking up raw text into discrete words or tokens so the model can process them more easily. Based on the whitespaces each word in the encrypted sentences gets tokenized. Whitespaces have been employed because, as has been noted, all foreign languages tokenize based on whitespaces, and encrypted data can resemble foreign languages. Thus, this path has been taken. The original words in the sentences are then represented by sequences of these integers created from tokenized text. In order to maintain uniformity and guarantee that all input sequences for model training are the same length, padding may also be used because sentences can differ in length. This tokenized and cleaned dataset becomes the foundation on which the word embeddings are built.

The data is provided for embedding training once it has been encrypted and tokenized. The following are the parameters of the Word embedding that are used:

Index	Context Pair Method	Min Count	Window Size	Vocab Size	Dimension
1	CBOW	5	3	1190075	256
2	Skip-Gram	5	3	1190075	256
3	Skip-Gram	8	5	1019028	256
4	Skip-Gram	15	4	766934	256
5	Skip-Gram	25	4	534352	256
6	Skip-Gram	35	4	421665	256
7	Skip-Gram	50	4	328481	256

Table 5.1: Wikipedia_512_Pretraining Corpus on Gensim Word2Vec

Lower minimum counts, like 5, capture a wider vocabulary, with over 1.19 million words while higher thresholds, like 50, focus on more common terms and may even exclude crucial words, reducing the vocabulary size to 328,481. This trade-off suggests a balance between capturing more words and retaining only those that appear frequently enough for robust representation. The window size, which indicates the number of context words that are taken into account surrounding a target word, stays largely constant at 4. A smaller window size concentrates on closer context whereas a larger one usually captures broader contextual relations.

With a vocabulary size of 421,665, the configuration, Min Count=35, employs a Skip-Gram model with a window size of 4, suggesting that it places more emphasis on frequent terms. A min count of less than 35 tends to make the vocabulary size unreasonably big. This specific setting aims to strike a balance between including enough contextual information while filtering out infrequent words that contribute to noise.

The Word2Vec model receives tokenized phrases once more. The Skip-Gram method will be employed by the model to acquire word representations. This entails using the current word to predict the words in the surrounding context. To improve the word embeddings, the model will run over the complete dataset several times (five epochs in this example). Following training, every word in the vocabulary has a corresponding vector in the embedding space that represents its context-based semantic meaning. The model can be saved for subsequent use after it has been trained. Depending on the demands of the task at hand, these embeddings are either left static or progressively refined during the neural network training process. The neural network's capacity to comprehend word meaning in context is improved by the pre-trained Word2Vec embeddings, which offer a solid foundation for capturing linguistic nuances.

Finally, for model training, a standard LSTM (Long Short-Term Memory) model has been used for binary text data classification. The LSTM (Long Short-Term Memory) model classifies text sequences using pre-trained Word2Vec embeddings, loaded into a fixed embedding_matrix to provide semantic word representations. An LSTM layer with 256 units sits after an embedding layer in the Sequential model structure, which enables it to capture long-term dependencies through gating methods. ReLU activation and decreasing unit density layers (256, 128, 64) enhance features for binary classification. The model employs the Adam optimiser with a low learning rate of 0.000005 for stable training and BinaryCrossentropy for calculating losses. Using a batch size of 128 for training, the learning rate is adjusted when progress slows down using ReduceLROnPlateau and EarlyStopping to avoid overfitting.

Again, pre-trained Word2Vec embeddings are used by the GRU (Gated Recurrent Unit) model to represent words and capture their semantic associations. The GRU makes use of this prior knowledge by loading these embeddings into a predefined embedding matrix. The model is composed of two GRU layers: the first, which has 256 units, and return_sequences=True, allows intermediate states to be accessed by the second, which has 128 units. This configuration is useful for tasks like sentiment analysis since it helps to capture sequential dependencies in text. Regularization is aided by dense layers and a Dropout layer with a rate of 0.3; probabilities for binary classification are output by the final sigmoid-activated layer. The model makes use of the RMSprop optimiser with a learning rate of 0.00005 and BinaryCrossentropy as the loss function. Using a batch size of 128 to balance speed and memory efficiency, training is optimized using EarlyStopping and ReduceLROnPlateau to prevent overfitting and dynamically modify learning rates. In [30] it is stated that RMSprop is adaptive in the learning process and has the capability to work with mini-batches. It leverages the exponentially weighted averages of the gradients to update its parameters and is normally considered a prominent choice for RNN-based models. This is why RMSprop has been utilized in this research. Again, binary cross-entropy as a loss function has been implemented as it is an ideal choice for the binary classification, as explained in section [30].

So, a saved embedding model has been imported which has been trained on encrypted data and for the embedding layer, word2vec model vectors are being used and being kept fixed without any change to the dataset used for the substitution dictionary. Once the embedding layer is trained, the next step is to train a neural network model LSTM or GRU. The tokenized and embedded data is fed into this model for further learning. Following model training, a thorough comparison is also performed using the F1-score, recall, and precision metrics, highlighting the advantages and disadvantages of each model. To gain a better understanding of the distribution of true positives, true negatives, false positives, and false negatives—as well as a deeper look into the categorization patterns of the model—confusion matrices are visualized for both models.

Accuracy, F1 score, precision, and recall are the evaluation criteria for both models. With a loss of 0.3678, the LSTM model yielded an accuracy of 83.60%, precision of 85.04%, recall of 85.89%, and F1 score of 85.46%. With an accuracy of 86.05%, F1 score of 87.16%, precision of 90.17%, recall of 84.34%, and a smaller loss of 0.3194, the GRU model outperformed the others by a small margin. Pre-trained Word2Vec embeddings are used by both models, and they remain fixed throughout training. The GRU model is organized with two stacked GRU layers, followed by dropout and dense layers for regularisation, whereas the LSTM model has a single LSTM layer with dense layers for refinement.

To summarise, in the proposed pipeline, the user first gives input. The input text goes through bigram substitution, tokenization, sequencing, and padding and is fed to the pre-saved RNN models LSTM, and GRU models to classify them into negative and positive labels. The tuple also carries the encrypted raw data (user input) using a bigram substitution cipher. The encryption is done without cleaning. This encryption is done in order to safeguard the user input from any Man-in-the-middle (MIM) attack. So, the tuple has only the label, and encrypted user input. The safety is ensured as the bigram cipher is based on Alphanumeric characters, special characters, and punctuation marks

which give a combination of 8890! which is challenging to crack using brute force attacks. Therefore, the transmission from sender to receiver is kept confidential while maintaining integrity.

Now, the receiver receives the label of the emotion. If the sender is in distress, the receiver will become aware of it as the emotion received through sentiment analysis will be labeled negative after being classified via the RNN models. Based on the label received, the authorities if the emergency text service platform will connect the user to the necessary channel, and if needed, will send reinforcements. To know exactly what the sender has texted, the receiver decrypts the encrypted raw data.

5.2 Potential Vulnerabilities

A side-channel attack (1) as depicted in figure 5.2, as opposed to an algorithmic cryptography vulnerability, takes advantage of data that has been disclosed from a cryptosystem as a result of flaws in its physical implementation [4]. Side-channel attacks can occur in electronic systems that use cryptographic keys and algorithms. Physical information leaks from the system, which might happen in software or hardware implementations, are the target of these attacks [3]. New attack vectors that cause information to leak during computation have resulted from this.

Inferring details about plaintext or secret keys can be accomplished by examining variations in power consumption during cryptographic procedures. The power consumption might disclose which substitutions are being accessed based on their memory address or cache hits/misses when the algorithm looks for the presence of a bigram in the substitution dictionary to determine if the cipher's implementation makes use of particular data structures (like arrays or hash tables). Using computer resources for the execution of encryption algorithms results in power consumption [12]. Distinct power consumption patterns will be displayed via various data structures. For instance, if the bigram substitution dictionary is implemented as an array, the power consumption of each index access may be comparatively constant. Accessing elements may use various amounts of power depending on the memory address being accessed and whether the storage substitution pair is stored in a dictionary, array, or linked list.

Again, in order to keep an eye on the target device's power usage, a little resistor can be inserted into the power supply line [5]. By using a resistor, an attacker can examine a single power consumption curve to gather statistical data and see changes in power usage that correspond with certain actions being carried out by the device [1]. To smooth out the power consumption curve, one potential approach is to place a capacitor across the smart card's power supply lines [1].

Apart from side-channel attacks, several other types of attacks target cryptographic algorithms. For example, given a scenario, Alice and Bob use a substitution cipher to exchange encrypted messages over a secure communication channel. But one of their enemies, Eve, intercepts their messages and tries to decipher them. The adversary is speculated to only have access to a set of ciphertexts in this sort of attack, defined as a ciphertext-only attack (2) as illustrated in figure 5.2. Frequency analysis is one of the best ways to decipher a substitution cipher. Usually, frequency analysis is the first step

in the manual deciphering of replacement ciphers [10]. For instance, Eve could surmise that a symbol appears frequently in the ciphertext and that it probably refers to one of the common letters in the English language [40]. Eve is able to eventually recreate the original plaintext by iteratively rewriting the symbols based on these frequency patterns in a methodical manner.

The letter frequency is the frequency of each letter occurring in all the texts calculated which helps determine the fundamental correspondence between the letters and lessens the difficulty of decryption [40]. Eve's efforts will, however, eventually be in vain because the encryption method used is bigram substitution-based. Eve uses bigram frequency analysis, which is a more advanced method. After that, Eve will switch to a more advanced strategy called bigram frequency analysis. Using this technique, pairings of letters known as bigrams (th, he, in, etc.)which frequently occur together in English—are analyzed.

If the attacker is unsuccessful, they will attempt a time-consuming brute force attack, attempting every key until they discover the one that decrypts the ciphertext into legible text. Brute force matches the frequency distribution of letters [13]. Larger key spacing makes it harder to guess the right key during brute-force attacks. A conventional substitution cipher generates a permutation of 26 letters by substituting any one of the alphabet's 26 letters. Despite its seemingly huge size, 26! is susceptible to frequency analysis because every letter in the ciphertext has a direct equivalent in the plaintext. On the other hand, bigram substitution normally uses 8836!, an astronomically enormous number. It would be practically difficult to brute-force this many combinations in a reasonable amount of time, even if computers were able to verify billions of keys each second. Furthermore, we have integrated special characters, punctuation, and letters contributing to a combination of 8836(!) factorial which is way more larger than 676!

Finally, by regularly altering the mappings of substitutions, the cipher's defenses against chosen-ciphertext attacks (3) mentioned in figure 5.2 at the receiver's end when an adversary has the ability to enter their ciphertext into the decryption process and inspect the outcome in plaintext and render brute-force attacks unfeasible because of the randomness of the substitutions can be strengthened. Even if an attacker could predict some of the mappings, they would have to deal with a lot of ciphertexts before seeing any significant trends because the system relies heavily on randomization (caused by shuffling). It gets harder for an attacker to carry out a good frequency analysis the more intricate and unpredictable the frequency distribution is.

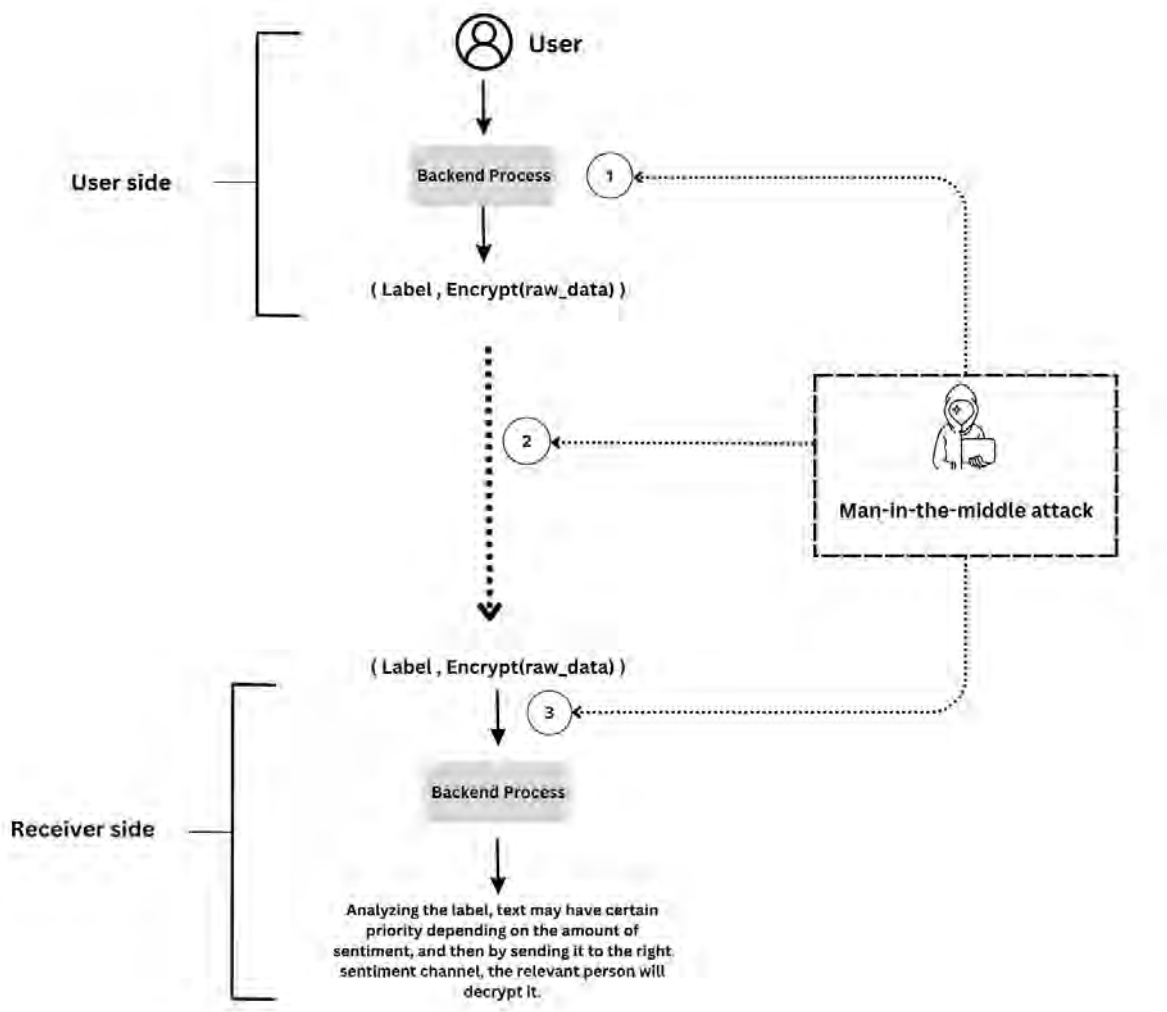

Figure 5.2: Susceptible point of attack

If the attacker is unsuccessful, they will attempt a time-consuming brute force attack, attempting every key until they discover the one that decrypts the ciphertext into legible text. A person applying a brute-force technique to decode using letter frequency analysis and pattern matching was one of the only algorithms that has ever been developed for the decryption of substitution-ciphered data [13]. Larger key spacing make it harder to guess the right key during brute-force attacks. A conventional substitution cypher generates a permutation of 26 letters by substituting any one of the alphabet's 26 letters. Despite its seemingly huge size, 26! is susceptible to frequency analysis because every letter in the ciphertext has a direct equivalent in the plaintext. Bigram substitution uses 8836! , an astronomically enormous number. It would be practically difficult to brute-force this many combinations in a reasonable amount of time, even if computers were able to verify billions of keys each second.

Finally, by regularly altering the mappings of substitutions, the cipher's defenses against chosen-ciphertext attacks and render brute-force attacks unfeasible because of the randomness of the substitutions can be strengthened. Even if an attacker could predict some of the mappings, they would have to deal with a lot of ciphertexts before seeing any significant trends because the system relies heavily on randomization (caused by shuffling).

It gets harder for an attacker to carry out a good frequency analysis the more intricate and unpredictable the frequency distribution is.

5.3 Evaluation

In summary, this report has identified the need for further research to improve the analysis of this proposed work. The proposed research objectives and work plan aim to build and evaluate a secure communication system employing substitution encryption.
To measure the effectiveness of the model, accuracy and F1 score are used. Accuracy describes the percentage of correct predictions made when all cases are considered. No meaning or context F1 score takes into account both precision and recall when gauging how well a model performs, hence the need for comprehensive judgment.

5.3.1 Performance Evaluation matrices

For evaluating the performances of the models, we used accuracy and F1 score.

a. **Accuracy:** Accuracy measures the correct percentages of the prediction (both true positive and true negative) out of all the predictions that have been made.

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

b. **F1 Score:** The F1 score is the balanced indicator of a model's performance. It is the harmonic mean of precision and recall. It is calculated as follows:

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

c. **Recall:** Recall evaluates the model's ability to capture all positives. We calculate the ratio of accurate positive predictions to positive facts.

$$\text{Recall} = \frac{TP}{TP + FN}$$

d. **Precision:** Every model positive prediction is precise. The percentage of genuine positive forecasts out of all positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

5.4 Limitations and Future Work

One of the key limitations of this research is the reliance on substitution encryption, which is known for its lack of robustness compared to more advanced encryption techniques. The substitution cipher is seen to be insufficiently secure for many practical uses, even if it offers a basis for investigating sentiment analysis on encrypted data. The study recognizes the need for more secure techniques by acknowledging the possibility of security breaches at various points of data transmission and processing. Semantic preservation also plays a major role in the accuracy of NLP models; this can be affected by lemmatization and the model's ability to correctly understand the top 10 semantically comparable terms. This restriction emphasizes the need to use stronger encryption techniques like Data Encryption Standard(DES), Blowfish. and Advanced Encryption Standard (AES). Additionally, by using public and private keys to provide more secure data sharing, an asymmetric encryption technique like RSA might be implemented to further improve data security. Furthermore, the research aims to incorporate homomorphic encryption in the future, allowing a comparison with our implementation. These constraints indicate that in order to achieve greater performance and security in future iterations of this research, which will be undertaken, a more complete encryption method combined with enhanced sequence-to-sequence models like transformers can be implemented.

Chapter 6

Result

6.1 Result

For this analysis, several natural language processing (NLP) techniques and models were utilized to investigate and demonstrate the interconnections of words in a large text data set. Word Embeddings, Word2Vec analogies and word similarity depiction techniques were applied to determine the relational aspect of language. The research further gauged the performance of Skip-Gram, Continuous Bag of Words (CBOW) methods as well as deep learning models such as LSTM and GRU in analyzing context and language structure.

6.1.1 Word Embeddings Similarity

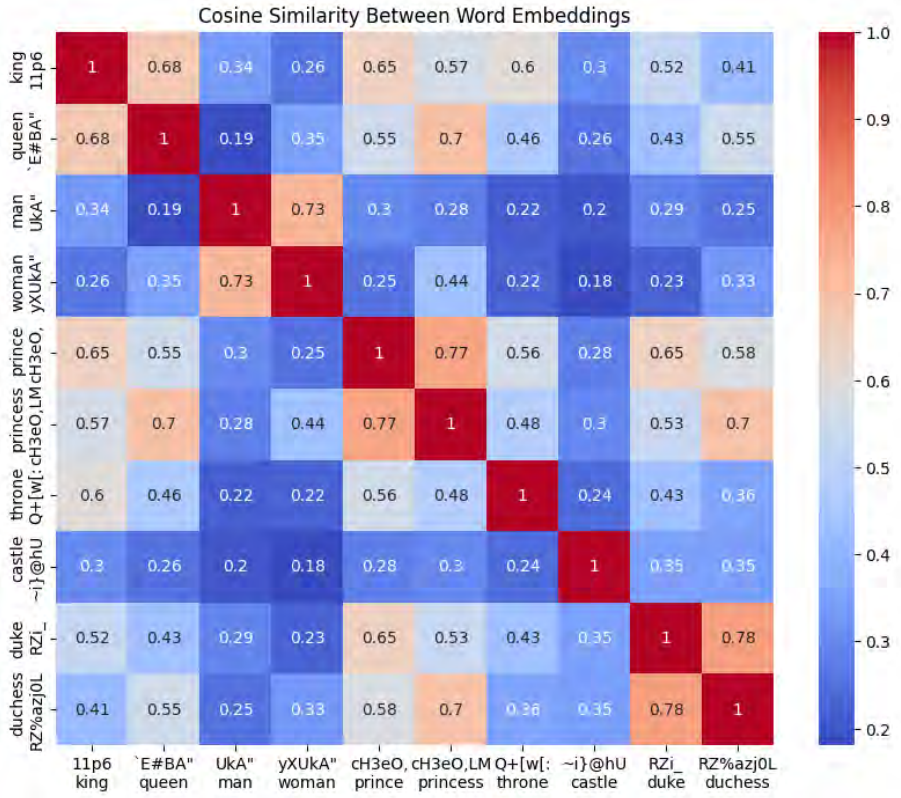

Figure 6.1: Cosine Similarity Between Word Embeddings

Figure 6.1 is a heatmap depicting the relationship between different word vectors pertaining to royalty and gender using the Word2Vec model. Cosine similarity is a concept used in determining the relationship between 2 words in vector space, in this case, how similar word vectors are to one another ranging from 0 to 1 where 1 means the two vectors are the same and 0 means the two vectors are entirely different. The heat map shows how close relations the words in the defined pairs have in the vector space.

At the two extreme edges, the x-y axes are occupied by men and women wording such as King, Queen, Prince, and Princess, while gender-neutral and site specific words including Castle and Throne are in between. The heatmap is designed in such a way that it contains various colors, with high similarity being represented in red while low similarity in blue. The diagonal of the matrix finds the similarity of each word compared with itself. This is what one would expect.It is usually observed that some of the strongest relationships exist in the closeness of word pairs. Like the case of expectation, a score of 0.77 Similarity score denotes the level of equality between the words princess and prince. They represent the different sexes of royalty. Another title that is similar to high degrees is the title of the combat outreach i.e. duke and duchess, which reached 0.78. Further, these scores demonstrate how well the model performs in explaining the connections between the titles which have gender distinctions but still form one cage linguistically.

Yet another word pair, king and queen, has a moderate score of 0.68 which again shows a pertinent relationship between the two words although it is more filled with contrasts than how it is with the gender interactive words prince and princess. For example, the words throne and castle have medium levels of similarity with both the royalty titles as well as with each other because these words have been observed to occur together in the same context often. This heatmap, in conclusion, empirically demonstrates the Word2Vec model's ability to capture the high degree of semantic relatedness between the specified words. Independently functioning counterpart words (prince/princess, duke/duchess) tend to group together within the space with high similarity scores while other terms which are regarded to be unrelated are easily separated and show low similarity.

6.1.2 Word2Vec Analogy

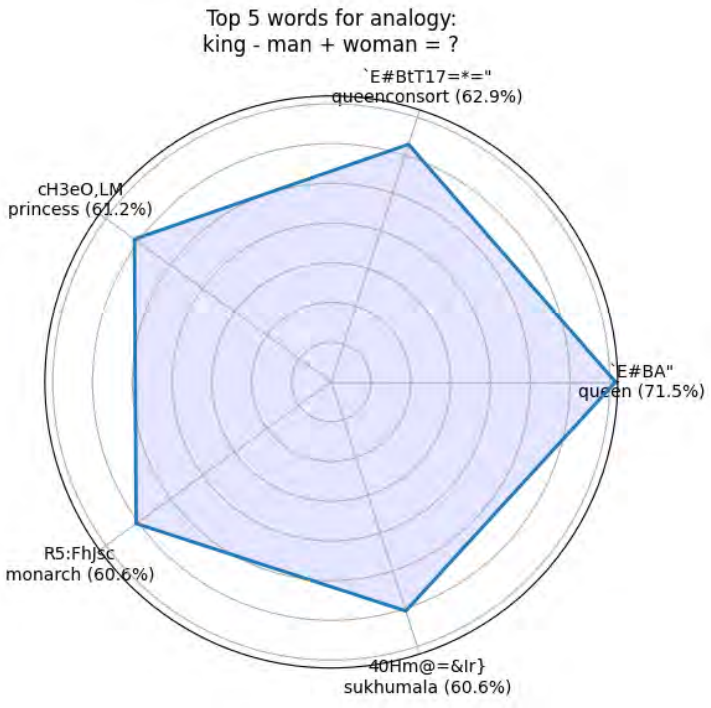

Figure 6.2: Top 5 Words Analogy when King-Man+Woman=?

Figure 6.2 contains a radar plot that shows the five most frequent words which have been obtained through the Word2Vec analogy king – man + woman = ?. The purpose of this analogy is to find a word which is most similar to the word 'Queen' in the context of removing the male-specific 'man' from 'king', and adding a female-specific 'woman' to it. With a similarity score of 71.5%, the model's best outcome, queen, matches our expectations of a female monarchy in the royal sphere. Related phrases that are semantically close to royalty and gender include princess (61.2%) and monarch (60.6%). Furthermore, terms like "sukhumala" (60.6%) and "queen consort" (62.9%) are noted, suggesting that the model accurately represents the complex functions connected to queenship. The above-mentioned radar chart shows the ability of the model to reason by analogy which in turn supports an understanding of gender and roles in the context of the model used which is compatible.

6.1.3 Word Similarity visualization

t-SNE: A statistical technique called t-distributed stochastic neighbor embedding (t-SNE) assigns a position to each datapoint in a two or three dimensional map in order to visualize high-dimensional data [48].

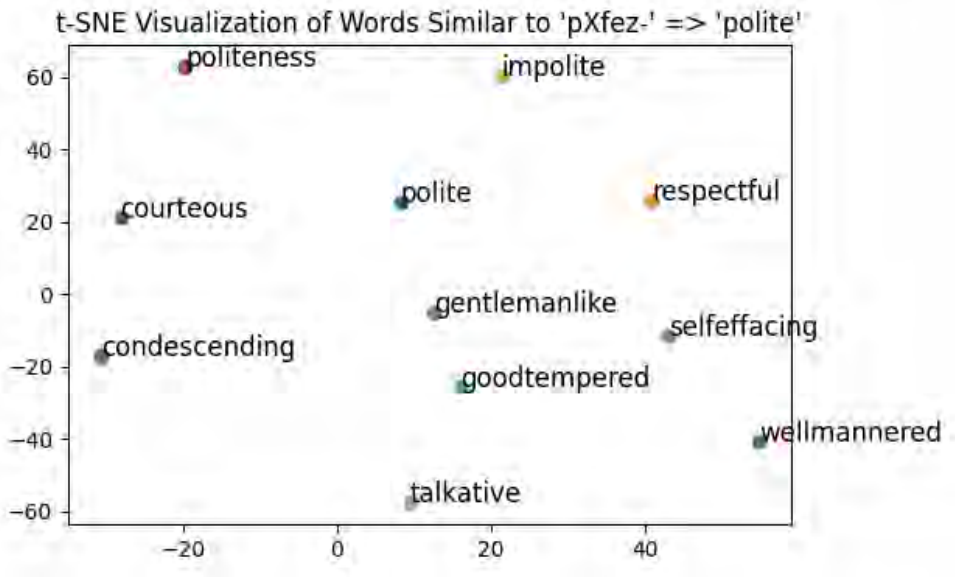

Figure 6.3: Word Similarity visualization(polite)

In Figure 6.3, the extension evaluates t-SNE in the context of the similarity of word embedding of words similar to 'polite' (encoded as 'pXfez-') from the Word2Vec model. The created 2D graph also contains clusters of terms of similar meaning. Polite, for example, courteous and respectful can be grouped together since they share the same context of a positive regard to people. Impolite and condescending are kept at a distance, further reinforcing their negative views. The distance between words indicates their likeness with respect to meaning, words far apart are not similar. This image gives a good example of how capable t-SNE is to discriminate all the nuances of meaning and the relationships between the words in embedding fortified embeddings.

Figure 6.4: t-SNE Word Cluster

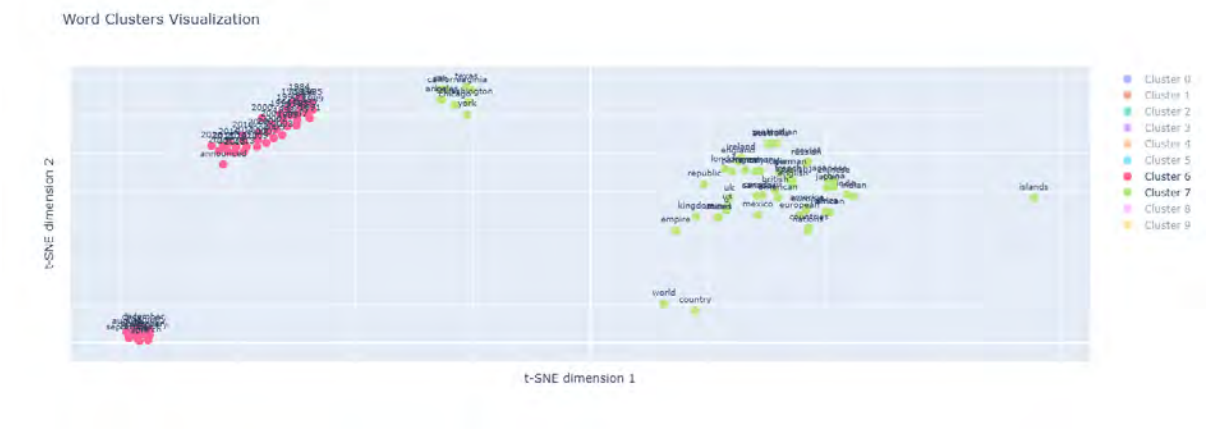

Figure 6.5: t-SNE Word Cluster

As observed in the first t-SNE diagram, the word clusters are even spread over the plot, indicating the presence of 10 different clusters. The overlapping clusters suggest word relationships are complex, where words from different clusters are in close proximity due to similarity in meaning. In this case, the clusters are not as well separated, which may mean that the model or data contains some level of more complex or even multidimensional relationships between groups of words. The t-SNE mechanism also performs two-dimensional drawing of word vector spaces, which introduces the problem of cluster overlapping as most interactions are non-linear, hence cocoons of some clusters remain obscured showing difficulty in understanding the different clusters even with color features.

Turning to the second t-SNE rendering, it is easy to note that the word clusters are less entangled which signifies that the semantic boundaries between the word groupings are much clearer. Clusters 7 (green) and 6 (pink) show a high degree of intra-cluster connectedness, that is, the terms in these two groups are very much related. The extent of the cluster boundaries implies that there is a clearer understanding of the data with less ambiguity when compared to the first diagram. The cause of this greater distances between the clusters may be better separation between the clusters or may be due to the use of different types of clustering features that yield clearer clusters. All in all, the t-SNE scatter plot serves the purpose in showing the differences in word embeddings and how the words are formed within the clusters.

6.1.4 Comparison between Skip-Gram and CBOW

Dataset	Wikipedia_512_Pretrainig Corpus							
	Skip-Gram				CBOW			
	Spearman		Pearson		Spearman		Pearson	
	Correlation	Pvalue	Correlation	Pvalue	Correlation	Pvalue	Correlation	Pvalue
simlex-999	0.3829	3.09^e-36	0.3942	1.73^e-38	0.3564	2.73^e-31	0.3801	1.07^e-35
wordsim-353	0.6987	2.21^e-50	0.6684	9.96^e-45	0.5916	5.10^e-33	0.5893	1.02^e-32
Google Analogies	Accuracy				Accuracy			
	0.6996				0.65			

Figure 6.6: Performance comparison between Skip-Gram and CBOW on simlex-999, wordsim-353

SimLex-999 [11] and WordSim-353 [45] serve as gold standards because they evaluate different aspects of semantic understanding. To evaluate how effectively models can capture genuine similarity, SimLex-999 focuses on differentiating between strict similarity (e.g., "car" and "automobile") and mere relatedness (e.g., "car" and "road"). On the other hand, WordSim-353 is more suited for general semantic tasks since it incorporates both relatedness and similarity.

A consistent pattern is seen by analyzing the Word2Vec model's performance on WordSim-353 and SimLex-999 using both Pearson and Spearman correlation metrics. The model gets a strong 0.6987 Spearman correlation and a little lower 0.6684 Pearson correlation for WordSim-353, both of which are highly significant given the low p-values. This indicates a robust alignment with human judgments of general semantic relatedness, suggesting that the model captures how words are generally correlated. On the other hand, SimLex-999 exhibits lower but statistically significant correlations (Pearman: 0.3829, Spearman: 0.3942), indicating an appropriate degree of agreement with human evaluations of strict semantic similarity. The slightly higher Spearman correlations between the two datasets suggest the possibility of some non-linear links, which are typical in language data when connections aren't strictly linear, between the model scores and human judgments.

Hence, while the model's lower SimLex-999 scores emphasize the challenge of capturing nuanced, fine-grained similarity, its stronger alignment with WordSim-353 implies that it performs better in capturing larger semantic correlations. The Google Analogies [44] test set is a state of the art dataset to evaluate semantical and syntactic similarity. The accuracy achieved in Skip Gram is 0.6996 and in CBOW is 0.65. As a result, both datasets provide complimentary insights that enable a thorough assessment of word embeddings.

6.1.5 LSTM

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
1	0.6931	0.4995	0.6859	0.6057
2	0.6815	0.6007	0.6638	0.6596
3	0.6498	0.6969	0.5845	0.7771
4	0.5611	0.7782	0.5063	0.7953
5	0.4963	0.8046	0.4756	0.8182
16	0.3845	0.8365	0.3907	0.8302
17	0.3818	0.8355	0.3860	0.8315
18	0.3770	0.8354	0.3730	0.8404
19	0.3793	0.8351	0.3834	0.8289
20	0.3765	0.8364	0.3807	0.8320

Table 6.1: Training and Validation Results over Epochs

After twenty training epochs, the LSTM model's accuracy increased gradually from 49.95% to 83.64% on the training set. Similarly, the validation accuracy increased from 60.57% to 83.20%, and the validation loss decreased in subsequent epochs.

Figure 6.7: LSTM

The model's learning progress and convergence behavior has been visualized using training and validation curves.

6.1.6 GRU

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
1	1.8324	0.6008	1.4248	0.7695
2	1.3221	0.7913	1.0837	0.8247
3	1.0310	0.8173	0.8595	0.8333
4	0.8257	0.8269	0.7204	0.8263
5	0.6751	0.8342	0.6003	0.8398
26	0.3186	0.8654	0.3258	0.8609
27	0.3168	0.8655	0.3095	0.8669
28	0.3134	0.8678	0.3257	0.8591
29	0.3112	0.8701	0.3142	0.8659
30	0.3033	0.8737	0.3021	0.8676

Table 6.2: Training and Validation Results over Epochs

After thirty training epochs, the GRU model's accuracy increased gradually from 60.80% to 87.37% on the training set. The training accuracy increased along with the validation accuracy from 78.95% to 86.76%, and the validation loss decreased in subsequent epochs.

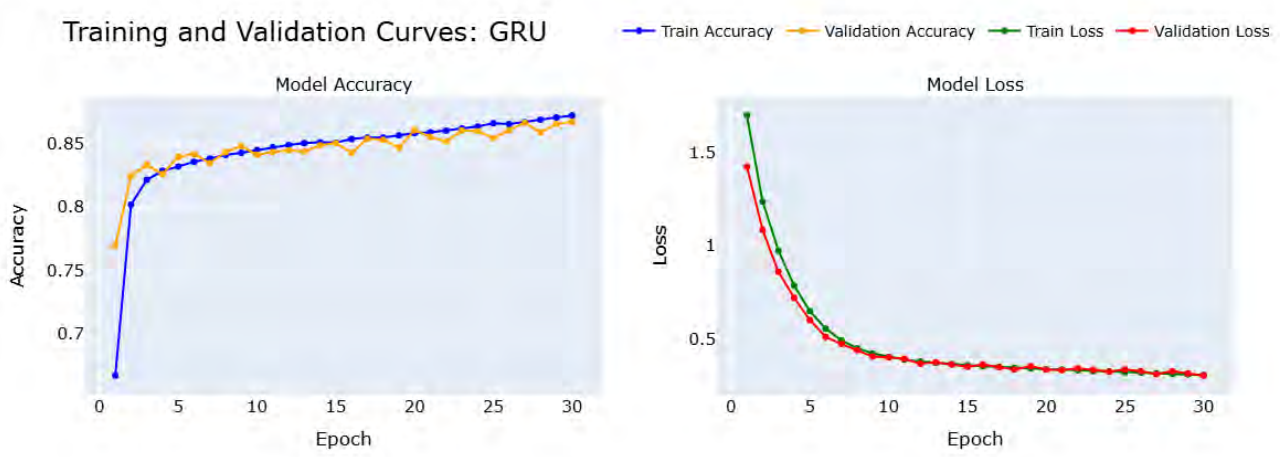

Figure 6.8: GRU

The model's learning progress and convergence behavior has been visualized using training and validation curves.

6.1.7 Comparison

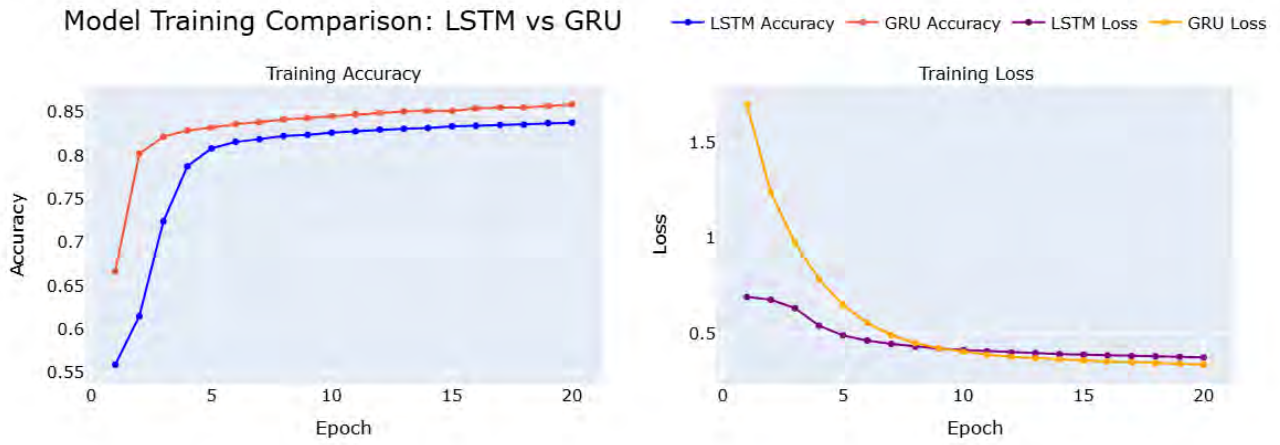

Figure 6.9: Comparison Between LSTM and GRU

Model	Accuracy	F1 Score	Precision	Recall	Loss
LSTM	83.47%	85.07%	86.28%	83.90%	0.3757
GRU	86.15%	87.59%	88.11%	87.07%	0.3133

Table 6.3: Model Performance Metrics

In terms of accuracy, GRU outperformed the LSTM, with the former achieving 86.15% while the latter achieved 83.47%. While GRU is better than LSTM in Recall (87.07% vs. 83.90%) Though there was just a slight difference in total performance, the GRU fared better in terms of precision (88.11% vs. 86.28%) and loss (0.3133 vs. 0.3757).

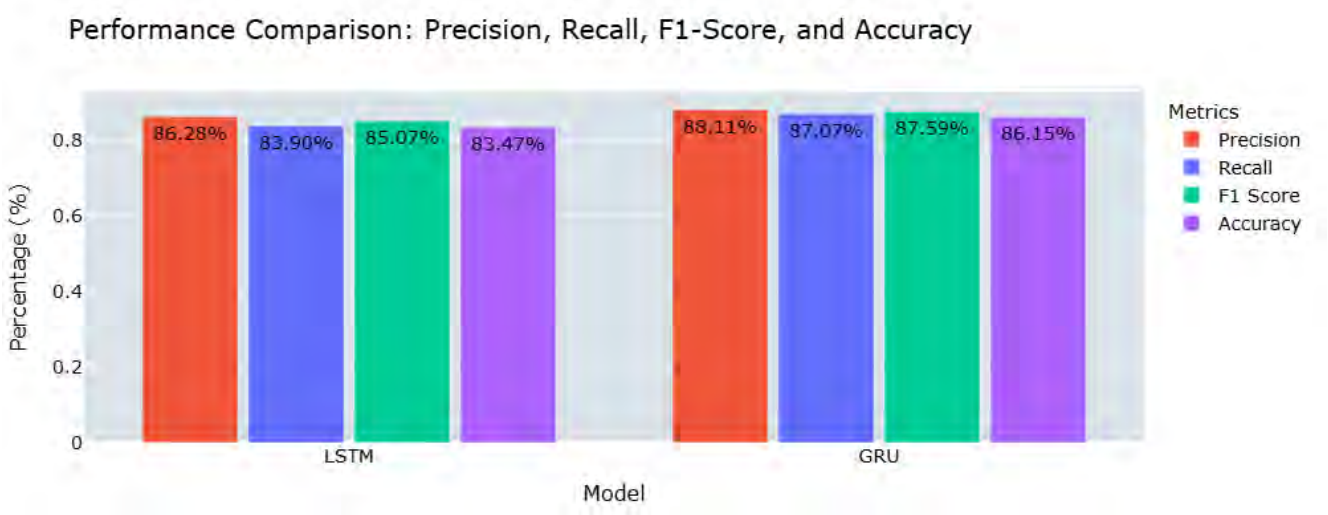

Figure 6.10: Precision, Recall, F1 score Accuracy, Comparison

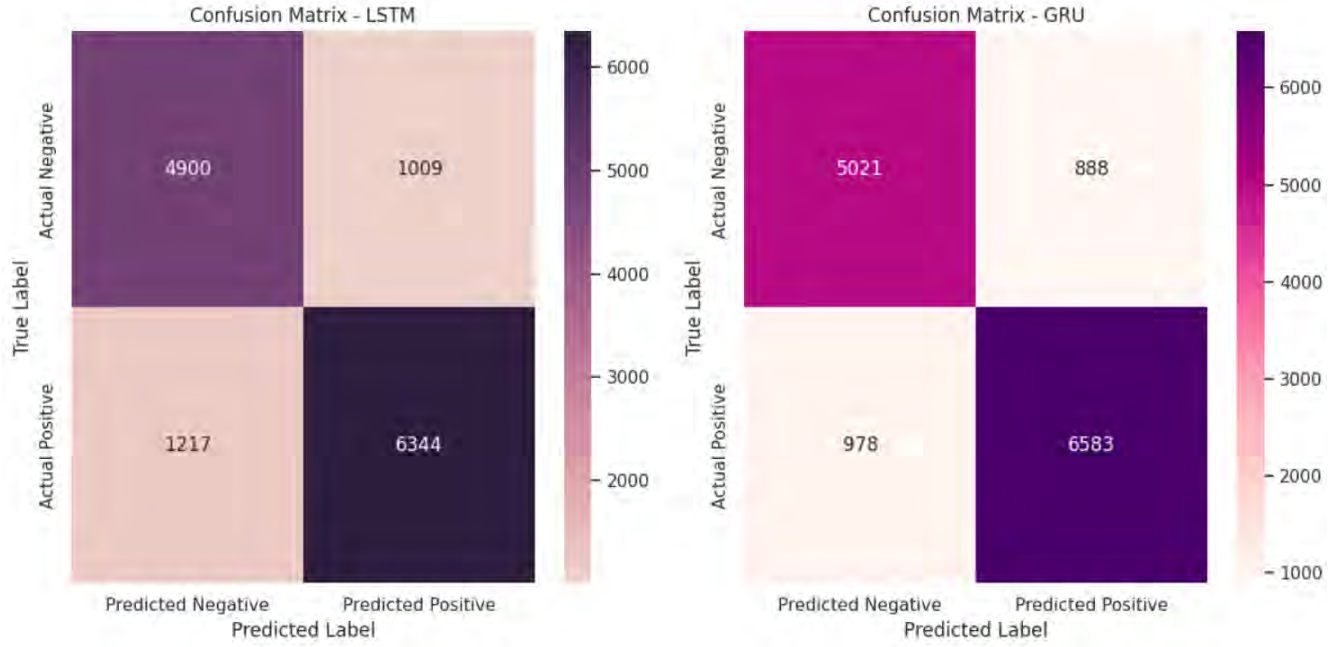

Figure 6.11: Confusion Matrix

The LSTM and GRU models' confusion matrices demonstrate that both models classify the data identically. In contrast to the GRU model, which accurately identified 5021 true negatives and 6583 genuine positives, the LSTM model correctly recognized 4900 true negatives and 6344 real positives. Compared to the LSTM, the GRU performed better with slightly fewer false positives (888 vs. 1009) and false negatives (978 vs. 1217).

From the analysis, it is evident that every NLP method or model has its advantages over the other in terms of word relationships, meaning, et cetera. Word2Vec analogy and similarity graphs did an excellent job at depicting language structures, while Skip-Gram and CBOW comparisons illustrated their differences in word prediction approaches.

Sequential data was well managed by LSTM and GRU techniques, with a difference in efficiency between the two. It can be concluded that all these methodologies are based on different principles and help in decoding the semantics of the words in the given language.

Chapter 7

Conclusion

In conclusion, this research delves into a relatively unexplored intersection between Natural Language Processing (NLP) and encryption, offering innovative insights into how encrypted text can be analyzed for sentiment detection without compromising privacy or data security. The research intends to evaluate the effectiveness of sentiment and emotional analysis on encrypted text by utilizing various approaches such as substitution ciphers, embedding, tokenization, and sequence padding. This innovative method not only protects sensitive textual data but also investigates whether models can reliably identify feelings and emotions even in the presence of encrypted input data.

By using substitution encryption, the proposed protocol makes sure that client and server communications are secure. It also makes use of sentiment analysis with LSTM and GRU models to assess the state of emotions. This technique ensures accuracy and the protection of confidentiality, which facilitates authorities' ability to react promptly to emergency alerts.

To provide a custom embedding, the word2vec embedding model is additionally trained using encrypted words. By more correctly capturing the syntactic and semantic meaning of words that have been learned on huge encrypted datasets, this technique aims to enhance the model's natural language processing (NLP) performance.

One of the main goals is to create a model that can process both encrypted and decrypted text. This kind of model can be very helpful in situations when strong data security is needed, like emergency response systems, where victim privacy is paramount. This research lays the groundwork for future investigations to create more reliable and safe sentiment analysis systems by comprehending the trade-offs and possible hazards of encryption in NLP applications.

Bibliography

[1] A. Shamir, "Protecting smart cards from passive power analysis with detached power supplies," in *Workshop on Cryptographic Hardware and Embedded Systems*, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:16855067.

[2] M. Atallah, C. McDonough, V. Raskin, and S. Nirenburg, "Natural language processing for information assurance and security: An overview and implementations," Feb. 2001. DOI: 10.1145/366173.366190.

[3] S. Guilley, P. Hoogvorst, R. Pacalet, and J. Schmidt, "Improving side-channel attacks by exploiting substitution boxes properties," Jan. 2007.

[4] A. Kaminsky, M. Kurdziel, and S. Radziszowski, "An overview of cryptanalysis research for the advanced encryption standard," Sep. 2010. DOI: 10.1109/MILCOM.2010.5680130.

[5] F.-X. Standaert, "Introduction to side-channel attacks," in Dec. 2010, pp. 27–42, ISBN: 978-0-387-71827-9. DOI: 10.1007/978-0-387-71829-3_2.

[6] X. Jing, Y. Hao, H. Fei, and Z. Li, "Text encryption algorithm based on natural language processing," in *2012 Fourth International Conference on Multimedia Information Networking and Security*, 2012, pp. 670–672. DOI: 10.1109/MINES.2012.216.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, 2013. arXiv: 1310.4546 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1310.4546.

[8] R. Socher, A. Perelygin, J. Wu, *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170.

[9] W. Y. Chong, B. Selvaretnam, and L.-K. Soon, "Natural language processing for sentiment analysis: An exploratory analysis on tweets," in *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, 2014, pp. 212–217. DOI: 10.1109/ICAIET.2014.43.

[10] B. Hauer, R. Hayward, and G. Kondrak, "Solving substitution ciphers with combined language models," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J. Tsujii and J. Hajic, Eds., Dublin City University and Association for Computational Linguistics, 2014, pp. 2314–2325.

[11] F. Hill, R. Reichart, and A. Korhonen, *Simlex-999: Evaluating semantic models with (genuine) similarity estimation*, 2014. arXiv: 1408.3456 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1408.3456.

[12] T. Nie, L. Zhou, and Z.-M. Lu, "Power evaluation methods for data encryption algorithms," *IET Software*, vol. 8, no. 1, pp. 12–18, 2014.

[13] A. Jain, R. Dedhia, and A. Patil, "Enhancing the security of caesar cipher substitution method using a randomized approach for more secure communication," *International Journal of Computer Applications*, vol. 129, pp. 6–11, Nov. 2015. DOI: 10.5120/ijca2015907062.

[14] M. Kanakaraj and R. M. R. Guddeti, "Nlp based sentiment analysis on twitter data using ensemble classifiers," in *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015, pp. 1–5. DOI: 10.1109/ICSCN.2015.7219856.

[15] E. Agrawal and P. Pal, "A secure and fast approach for encryption and decryption of message communication," *International Journal of Engineering Science and Computing*, vol. 7, p. 5, May 2017.

[16] K. Keerthi and B. Surendiran, "Elliptic curve cryptography for secured text encryption," in *2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)*, 2017, pp. 1–5. DOI: 10.1109/ICCPCT.2017.8074210.

[17] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2017, pp. 43–48. DOI: 10.1109/ISI.2017.8004872.

[18] A. Kushwaha, H. Sharma, and D. A. Ambhaikar, "Selective encryption using natural language processing for text data in mobile ad hoc network," in Jan. 2018, pp. 15–26, ISBN: 978-3-319-70541-5. DOI: 10.1007/978-3-319-70542-2_2.

[19] A. Naik, A. Saksena, K. Mudliar, A. Kazi, P. Sukhija, and R. Pawar, "Secure complaint bot using onion routing algorithm concealing identities to increase effectiveness of complain bot," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 1777–1780. DOI: 10.1109/ICECA.2018.8474839.

[20] Y. su and C. Kuo, "On extended long short-term memory and dependent bidirectional recurrent neural network," *Neurocomputing*, vol. 356, Feb. 2018. DOI: 10.1016/j.neucom.2019.04.044.

[21] B. Gaind, V. Syal, and S. Padgalwar, *Emotion detection and analysis on social media*, 2019. arXiv: 1901.08458 [cs.SI].

[22] O. Omolara and A. Jantan, "Modified honey encryption scheme for encoding natural language message," *International Journal of Electrical and Computer Engineering*, vol. Volume 9, pp. 1871–1878, Apr. 2019. DOI: 10.11591/ijece.v9i3.pp1871-1878.

[23] M. N. Sadat, M. M. A. Aziz, N. Mohammed, S. Pakhomov, H. Liu, and X. Jiang, "A privacy-preserving distributed filtering framework for nlp artifacts," *BMC Medical Informatics and Decision Making*, vol. 19, Sep. 2019. DOI: 10.1186/s12911-019-0867-z. (visited on 02/25/2023).

[24] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of depression-related posts in reddit social media forum," *IEEE Access*, vol. 7, pp. 44 883–44 893, 2019. DOI: 10.1109/ACCESS.2019.2909180.

[25] W. Ying, R. Xiang, and Q. Lu, "Improving multi-label emotion classification by integrating both general and domain-specific knowledge," *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 2019. DOI: 10.18653/v1/d19-5541.

[26] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, vol. 2, no. 7, 2020. DOI: 10.1002/eng2.12189.

[27] A. A. Badawi, L. Hoang, C. F. Mun, K. Laine, and K. M. M. Aung, "Privft: Private and fast text classification with homomorphic encryption," *IEEE Access*, vol. 8, pp. 226 544–226 556, 2020. DOI: 10.1109/ACCESS.2020.3045465.

[28] D. Biswas, "Privacy preserving chatbot conversations," in *2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2020, pp. 179–182. DOI: 10.1109/AIKE48582.2020.00035.

[29] Q. Feng, D. He, Z. Liu, H. Wang, and K.-K. R. Choo, "Securenlp: A system for multi-party privacy-preserving natural language processing," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3709–3721, 2020. DOI: 10.1109/TIFS.2020.2997134.

[30] Z. Hameed and B. Garcia-Zapirain, "Sentiment classification using a single-layered bilstm model," *IEEE Access*, vol. 8, pp. 73 992–74 001, 2020. DOI: 10.1109/ACCESS.2020.2988550.

[31] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets," *IEEE Access*, vol. 8, pp. 181 074–181 090, 2020. DOI: 10.1109/access.2020.3027350.

[32] K. Panchal, *Differential privacy and natural language processing to generate contextually similar decoy messages in honey encryption scheme*, 2020. arXiv: 2010.15985 [cs.CR].

[33] R. Podschwadt and D. Takabi, "Classification of encrypted word embeddings using recurrent neural networks," in *PrivateNLP@WSDM*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212727458.

[34] B. Saju, S. Jose, and A. Antony, "Comprehensive study on sentiment analysis: Types, approaches, recent applications, tools and apis," in *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, 2020, pp. 186–193. DOI: 10.1109/ACCTHPA49271.2020.9213209.

[35] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: A review of bert-based approaches," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 5789–5829, 2021. DOI: 10.1007/s10462-021-09958-2.

[36] D. Mahendran, C. Luo, and B. T. Mcinnes, "Review: Privacy-preservation in the context of natural language processing," *IEEE Access*, vol. 9, pp. 147 600–147 612, 2021. DOI: 10.1109/ACCESS.2021.3124163.

[37] T. Sosea and C. Caragea, "Emlm: A new pre-training objective for emotion related tasks," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021. DOI: 10.18653/v1/2021.acl-short.38.

[38] Q. Wang, W. Li, and Z. Jin, "Review of text classification in deep learning," *OALib*, vol. 08, pp. 1–8, Jan. 2021. DOI: 10.4236/oalib.1107175.

[39] D. Kim, G. Lee, and S. Oh, "Toward privacy-preserving text embedding similarity with homomorphic encryption," *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, 2022. DOI: 10.18653/v1/2022.finnlp-1.4.

[40] Y. Pan, "The scope of application of letter frequency analysis in substitution cipher," *Journal of Physics: Conference Series*, vol. 2386, p. 012 015, Dec. 2022. DOI: 10.1088/1742-6596/2386/1/012015.

[41] M. Wankhade, A. C. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022. DOI: 10.1007/s10462-022-10144-1.

[42] S. Poria, D. Hazarika, N. Majumder, and R. Mihalcea, "Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research," *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 108–132, 2023. DOI: 10.1109/TAFFC.2020.3038167.

[43] S. Stevens and Y. Su, *Memorization for good: Encryption with autoregressive language models*, 2023. arXiv: 2305.10445 [cs.CL].

[44] ACLWiki, "Google analogy test set (state of the art)," *ACLWiki*, 2024. [Online]. Available: https://aclweb.org/aclwiki/Google_analogy_test_set_(State_of_the_art).

[45] ACLWiki, "Wordsimilarity-353 test collection (state of the art)," *ACLWiki*, 2024. [Online]. Available: https://www.aclweb.org/aclwiki/WordSimilarity-353_Test_Collection_(State_of_the_art).

[46] D. Držík and J. Kapusta, *Effect of dimension size and window size on word embedding in classification tasks*, Jun. 2024. DOI: 10.21203/rs.3.rs-4532901/v1.

[47] A. Mishra, M. Li, and S. Deo, "Sentinellms: Encrypted input adaptation and fine-tuning of language models for private and secure inference," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 21 403–21 411, Mar. 2024. DOI: 10.1609/aaai.v38i19.30136. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/30136 (visited on 05/24/2024).

[48] [Online]. Available: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[49] L. D. Liello, $Lucadiliello/wikipedia_512_pretrainingdatasetsathuggingface$. [Online]. Available: https://huggingface.co/datasets/lucadiliello/wikipedia_512_pretraining.