Mixed-Code Sentiment Mining on Financial Social Media Interactions

BY

Rahul Kumar
(Admission No. 22MT0286)

Dissertation

SUBMITTED TO
INDIAN INSTITUTE OF TECHNOLOGY
(INDIAN SCHOOL OF MINES) DHANBAD

For the award of the degree of

MASTER OF TECHNOLOGY

MAY, 2024

कम्पयूटर विज्ञान एवं अभियांत्रिकी विभाग
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
भारतीय प्रौद्योगिकी संस्थान (भारतीय खनि विद्यापीठ), धनबाद
धनबाद-826004, झारखण्ड, भारत
INDIAN INSTITUTE OF TECHNOLOGY (INDIAN SCHOOL OF MINES), DHANBAD
DHANBAD-826004, JHARKHAND, INDIA

CERTIFICATE

This is to certify that the Dissertation entitled **Mixed-Code Sentiment Mining on Financial Social Media Interactions**, being submitted to the Indian Institute of Technology(Indian School of Mines), Dhanbad, by Mr. **Rahul Kumar**, Admission No **22MT0286** for the award of Degree of **Master of Technology** from IIT(ISM), Dhanbad, is a bonafide work carried out by her, in the Department of Computer Science and Engineering, IIT(ISM), Dhanbad, under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission. The results embodied in this dissertation have not been submitted to any other university or institute for the award of any degree or diploma.

Prof. Gadadhar Sahoo
Visiting Professor
Department of Computer Science and Engineering
Indian Institute of Technology(ISM) Dhanbad

कम्पयूटर विज्ञान एवं अभियांत्रिकी विभाग
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
भारतीय प्रौद्योगिकी संस्थान (भारतीय खनि विद्यापीठ), धनबाद
धनबाद-826004, झारखण्ड, भारत
INDIAN INSTITUTE OF TECHNOLOGY (INDIAN SCHOOL OF MINES), DHANBAD
DHANBAD-826004, JHARKHAND, INDIA

DECLARATION

I hereby declare that the work which is being presented in this dissertation entitled "**Mixed-Code Sentiment Mining on Financial Social Media Interactions**" in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Engineering** is an authentic record of my own work carried out during the period from **August 2023** to **April 2024** under the supervision of **Prof. Gadadhar Sahoo**, Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India.

I acknowledge that I have read and understood the UGC (Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions) Regulations, 2018. These Regulations were published in the Indian Official Gazette on 31st July, 2018.

I confirm that this Dissertation has been checked for plagiarism using the online plagiarism checking software provided by the Institute. At the end of the Dissertation, a copy of the summary report demonstrating similarities in content and its potential source (if any) generated online using plagiarism checking software is enclosed. I herewith confirm that the Dissertation has less than 10 % similarity according to the plagiarism checking software's report and meets the MoE/UGC Regulations as well as the Institute's rules for plagiarism.

I further declare that no portion of the dissertation or its data will be published without the Institute's or Guide's permission. I have not previously applied for any other degree or award using the topics and findings described in my dissertation.

Rahul Kumar
Admission No.: 22MT0286
M.Tech CSE
Department of Computer Science and Engineering
Indian Institute of Technology(ISM), Dhanbad

कम्पयूटर विज्ञान एवं अभियांत्रिकी विभाग
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
भारतीय प्रौद्योगिकी संस्थान (भारतीय खनि विद्यापीठ), धनबाद
धनबाद-826004, झारखण्ड, भारत
INDIAN INSTITUTE OF TECHNOLOGY (INDIAN SCHOOL OF MINES), DHANBAD
DHANBAD-826004, JHARKHAND, INDIA

CERTIFICATE FOR CLASSIFIED DATA

This is to certify that the Dissertation entitled **Mixed-Code Sentiment Mining on Financial Social Media Interactions** being submitted to the Indian Institute of Technology (Indian School of Mines), Dhanbad by **Mr. Rahul Kumar** for award of Master Degree in **Computer Science and Technology** does not contain any classified information. This work is original and has not yet been submitted to any institution or university for the award of any degree.

Signature of the guide(s)
Name: Prof. Gadadhar Sahoo
Date: **10 May 2024**

Signature of the Student
Name: Rahul Kumar
Date: **10 May 2024**

Phone: +91-326-2235273. Fax: +91-326-2296563. Website:
https://www.iitism.ac.in/index.php/computer_science

कम्पयूटर विज्ञान एवं अभियांत्रिकी विभाग
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
भारतीय प्रौद्योगिकी संस्थान (भारतीय खनि विद्यापीठ), धनबाद
धनबाद-826004, झारखण्ड, भारत
INDIAN INSTITUTE OF TECHNOLOGY (INDIAN SCHOOL OF MINES), DHANBAD
DHANBAD-826004, JHARKHAND, INDIA

CERTIFICATE REGARDING ENGLISH CHECKING

This is to certify that the Dissertation entitled "**Mixed-Code Sentiment Mining on Financial Social Media Interactions**" being submitted to the Indian Institute of Technology (Indian School of Mines), Dhanbad by **Mr Rahul Kumar** , Admission No **22MT0286**, for the award of Master of Technology has been thoroughly checked for quality of English and logical sequencing of topics.

It is hereby certified that the standard of English is good and that grammar and typos have been thoroughly checked.

Signature of Guide(s)
Name: Prof. Gadadhar Sahoo
Date: 10 May 2024

Signature of Student
Name: Rahul Kumar
Date: 10 May 2024

Phone: +91-326-2235273. Fax: +91-326-2296563. Website:
https://www.iitism.ac.in/index.php/computer_science

कम्पयूटर विज्ञान एवं अभियांत्रिकी विभाग
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
भारतीय प्रौद्योगिकी संस्थान (भारतीय खनि विद्यापीठ), धनबाद
धनबाद-826004, झारखण्ड, भारत
INDIAN INSTITUTE OF TECHNOLOGY (INDIAN SCHOOL OF MINES), DHANBAD
DHANBAD-826004, JHARKHAND, INDIA

COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IIT (ISM), Dhanbad and must accompany any such material in order to be published by the ISM. Please read the form carefully and keep a copy for your files.

TITLE OF DISSERTATION: Mixed-Code Sentiment Mining on Financial Social Media Interactions

AUTHOR'S NAME AND ADDRESS: Rahul Kumar

Rajeev Nagar
Patna
Bihar, 800024

COPYRIGHT TRANSFER

1. The undersigned hereby assigns to the Indian Institute of Technology (Indian School of Mines), Dhanbad, all rights under copyright that may exist in and to: (a) the above Work, including any revised or expanded derivative works submitted to the ISM by the undersigned based on the work; and (b) any associated written or multimedia components or other enhancements accompanying the work.

CONSENT AND RELEASE

2. In the event the undersigned makes a presentation based upon the work at a conference hosted or sponsored in whole or in part by the IIT (ISM) Dhanbad, the undersigned, in consideration for his/her participation in the conference, hereby grants the ISM the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive; in any format or medium, whether now known or

hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IIT(ISM) Dhanbad and live or recorded broadcast of the Presentation during or after the conference.

3. In connection with the permission granted in Section 2, the undersigned hereby grants IIT (ISM) Dhanbad the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice, and biographical information as part of the advertisement, distribution, and sale of products incorporating the Work or Presentation, and releases IIT (ISM) Dhanbad from any claim based on the right of privacy or publicity.

4. The undersigned hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data, or other material from the works of others, the undersigned has obtained any necessary permissions. Where necessary, the undersigned has obtained all third-party permissions and consents to grant the license above and has provided copies of such permissions and consents to IIT (ISM) Dhanbad.

GENERAL TERMS

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IIT (ISM) Dhanbad from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IIT (ISM) Dhanbad or is withdrawn by the author(s) before acceptance by the IIT(ISM) Dhanbad, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IIT(ISM) Dhanbad will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as an authorized agent for the others.

_____

Signature of the Student

ACKNOWLEDGEMENTS

It is indeed a great pleasure to express my sincere thanks to my guide **Prof. Gadadhar Sahoo**, Visiting Professor, Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, for his continuous support and guidance in this thesis. He was always there to listen and encourage me. He showed me different ways to approach a research problem and the need to be persistent to accomplish my goal. He inspired me to think through the problems and generate new idea. I will always be grateful to him for his valuable supervision and guidance.

I would like to thank **Dr. Chiranjeev Kumar** (Head of Department) and all the faculties of Computer Science and engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, for their valuable support and suggestions towards the research work.

I am grateful to my family members and friends for their constant support and motivation.

Last, but not the least, I want to thank my parents, for giving me the life in the first place and for educating me. I would like to thank my family for their unconditional support and encouragement to pursue my interest. It is a pleasure to express my deepest gratitude to all those who inspired me for the successful completion of the project.

Rahul Kumar
22MT0286
M.Tech. CSE
IIT ISM, Dhanbad
Date: 10 May 2024

ABSTRACT

This thesis presents an innovative approach to sentiment analysis of mixed-code (Hinglish) texts sourced from financial social media platforms. With the prevalence of code-mixing in online discourse, there arises an imperative for sophisticated sentiment analysis models capable of processing such multilingual data effectively. This study integrates GloVe (Global Vectors for Word Representation) word embedding methodologies with a Siamese neural network architecture, employing triplet loss to tackle the intricacies of sentiment analysis in a code-mixed environment where texts comprise a fusion of Hindi and English.

The methodology encompasses data collection, preprocessing, and exploratory analysis of a dataset comprising labeled Hinglish financial social media texts with sentiment polarity annotations. The neural network model leverages GloVe word embeddings and long short-term memory (LSTM) layers to capture both semantic nuances and contextual cues. Extensive experimentation and evaluation entail performance assessments, comparative analyses against existing methodologies, and the utilization of visualization techniques such as t-SNE (t-Distributed Stochastic Neighbor Embedding).

The findings underscore the efficacy of the proposed model in accurately discerning the sentiment of mixed-code financial texts, outperforming baseline approaches. This research contributes significantly to sentiment analysis by furnishing a robust solution tailored to handle code-mixed data, with promising applications in financial analysis, social media monitoring, and opinion mining.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Overview of Sentiment Mining

Sentiment analysis, often referred to as opinion mining, stands as a burgeoning domain within natural language processing (NLP), devoted to comprehending individuals' sentiments, evaluations, attitudes, and emotions regarding diverse entities and their characteristics. It encompasses a broad spectrum of tasks, including analyzing opinions about products, services, organizations, individuals, events, and topics, with a focus on discerning positive or negative sentiments.

An opinion can be expressed as a quintuple, $(e_i, a_{ij}, h_k, t_l, s_{ijkl})$, where e_i represents an entity, a_{ij} an aspect of entity e_i, h_k denotes the opinion holder, t_l signifies the time of giving the opinion, and s_{ijkl} indicates the sentiment of the opinion expressed.

This abstraction provides us with a structured approach to understanding the sentiment analysis problem, which comprises a wide range of interconnected subproblems. By defining and organizing the elements of opinions, we can better comprehend the complexity inherent in natural language data.

The emergence of sentiment analysis as a prominent research area can be attributed to several factors. Firstly, its myriad applications across diverse domains have spurred significant interest, particularly in the commercial sector, where sentiment analysis tools are in high demand. Secondly, the field presents numerous challenging research problems that were previously unexplored, driving researchers to innovate novel solutions.

Crucially, the proliferation of opinionated data on social media platforms has propelled sentiment analysis to the forefront of research. The abundance of user-

generated content provides an unprecedented resource for studying human opinions at scale, fostering new avenues for investigation and analysis. As such, sentiment analysis holds significance for NLP and intersects with disciplines such as management sciences, political science, economics, and social sciences, where understanding public sentiment is paramount.

In this thesis, we delve into sentiment analysis in financial social media interactions, focusing on the challenges posed by mixed-code text. Leveraging deep learning techniques, including LSTM and Siamese architecture, we aim to develop robust models capable of effectively mining sentiments from diverse linguistic expressions prevalent in financial discourse on social media platforms. Through empirical analysis and discussion, we seek to advance the understanding and application of sentiment analysis methodologies in real-world contexts.

1.2 Need of Sentiment Mining

In the contemporary interconnected global landscape characterized by the swift dissemination of information through digital mediums, the exigency for sentiment mining has attained unprecedented prominence. A profound comprehension of public sentiment is a pivotal imperative for enterprises, institutions, and policymakers alike, facilitating judicious decision-making and adept adjustment to market dynamics.

1. **Business Insights**: Sentiment mining is an invaluable conduit for discerning consumer perceptions and preferences, empowering enterprises to gauge customer satisfaction, discern emerging trends, and tailor offerings accordingly. Through monitoring sentiment across social media platforms, organizations proactively address customer concerns, safeguard brand reputation, and capitalize on burgeoning market opportunities.

2. **Market Analysis**: Within financial domains, sentiment mining assumes a paramount role in prognosticating market trajectories, evaluating investor sentiment, and mitigating risks. Through meticulous scrutiny of sentiment cues derived from social media dialogues, news sources, and financial disclosures, traders and investors navigate markets with data-driven acumen, positioning themselves ahead of market shifts.

3. **Brand Management**: Preserving a positive brand image stands as a quintessential objective for entities spanning diverse industries. Leveraging sentiment mining,

organizations surveil online discourse surrounding their brand and competitors, identifying potential public relations exigencies and formulating strategies to bolster brand perception and foster enduring customer allegiance.

4. **Public Opinion and Policy Making**: Sentiment mining engenders profound implications for public policy formulation and governance. By dissecting public sentiment on social and political issues, policymakers discern prevailing public opinion, anticipate societal disquietudes, and adeptly craft responsive policies that resonate with societal exigencies.

5. **Healthcare and Public Health**: Within healthcare realms, sentiment mining furnishes indispensable aids in monitoring public health trends, surveilling disease outbreaks, and pinpointing avenues for healthcare service enhancements. By scrutinizing sentiment manifestations in online forums and social media spheres, healthcare professionals glean insights into patient experiences, satisfaction levels, and burgeoning health exigencies.

In essence, sentiment mining constitutes an indispensable instrumentality for distilling actionable insights from the expansive reservoirs of unstructured data proliferating in the contemporary digital milieu. Through the adept harnessing of advanced analytical methodologies, organizations, and policymakers wield sentiment analysis as a potent lever for steering strategic decision-making, enriching customer experiences, and effectually addressing societal exigencies.

1.3 Unraveling Sentiments in Multilingual Financial Social Media Conversations

In today's digital age, financial social media platforms have emerged as vital hubs for investors, traders, and financial enthusiasts to exchange insights, opinions, and sentiments regarding various financial assets and market trends. These platforms, such as Twitter, StockTwits, and Reddit's financial forums, host a plethora of discussions, analyses, and updates that can significantly influence market sentiment and investment decisions. Amidst this digital discourse, a notable phenomenon arises: the prevalence of mixed-code interactions, where users seamlessly integrate multiple languages or dialects within their messages.

The need for mixed-code sentiment mining in financial social media text inter-

actions stems from the inherent complexity of these digital conversations. Unlike traditional textual data, financial social media text often incorporates a blend of languages, slang, abbreviations, and financial jargon, reflecting the diverse linguistic backgrounds and preferences of users. Consequently, sentiment analysis techniques designed for monolingual text may fall short in accurately capturing the nuanced sentiment expressed in mixed-code interactions.

Moreover, the dynamic and fast-paced nature of financial markets demands timely and accurate sentiment analysis to guide investment strategies and risk management decisions. Mixed-code sentiment mining presents a unique challenge and opportunity in this context, as it requires sophisticated natural language processing (NLP) techniques capable of deciphering sentiment across language boundaries and cultural nuances.

By delving into mixed-code sentiment mining on financial social media text interactions, researchers and practitioners aim to unlock valuable insights into market sentiment, investor sentiment, and emerging trends. Such insights can inform trading strategies, sentiment-aware investment models, and risk assessment frameworks, empowering market participants to make informed decisions in a rapidly evolving financial landscape. Thus, the exploration of mixed-code sentiment mining represents a crucial endeavor at the intersection of NLP, finance, and social media analytics, with profound implications for the understanding and interpretation of digital financial discourse.

1.4 Problem Statement

The primary goal of this study is to create a sentiment analysis model that can effectively understand and analyze financial social media content written in Hinglish. The proposed model aims to tackle challenges such as code-mixing, specialized terminology, and contextual ambiguity to generate reliable sentiment predictions. The objective is to develop a robust sentiment analysis framework capable of accurately interpreting sentiment in multilingual financial social media conversations.

1.5 Motivation Of Research

In today's digital era, the proliferation of social media platforms has transformed the landscape of communication, enabling individuals worldwide to express their opinions, sentiments, and emotions with unprecedented ease and immediacy. Within the realm of financial discourse, social media has emerged as a rich source of information, reflecting the collective sentiments and perceptions of market participants, investors, and consumers.

Despite the vast potential offered by social media data for understanding market dynamics and consumer behavior, extracting actionable insights remains a formidable challenge. This challenge is compounded by the linguistic diversity inherent in social media conversations, particularly in regions where code-mixing, the blending of multiple languages, is prevalent. In the context of India, where Hinglish—a fusion of Hindi and English—is commonly used in online interactions, the complexities of mixed-language text pose unique hurdles for sentiment analysis.

The motivation behind this research stems from the recognition of the critical need for sophisticated analytical tools capable of effectively mining sentiment from Hinglish financial social media text. Traditional sentiment analysis techniques often falter in handling code-mixed content, leading to inaccurate or biased results. By developing a specialized sentiment analysis model tailored to the nuances of Hinglish discourse, this research endeavors to bridge this gap and unlock valuable insights from the wealth of data available on social media platforms.

Moreover, the significance of this research extends beyond academic inquiry, resonating with practical implications for various stakeholders in the financial ecosystem. For businesses and marketers, the ability to decipher sentiment in Hinglish social media conversations offers a means to better understand consumer preferences, anticipate market trends, and enhance brand engagement strategies. Similarly, policymakers and regulators stand to benefit from insights derived from social media sentiment analysis, enabling them to formulate informed policies and respond effectively to public sentiment on financial matters.

By addressing these pressing challenges and harnessing the power of deep learning techniques, such as LSTM and Siamese architecture, this research endeavors to contribute to the advancement of sentiment analysis methodologies in the context of multilingual financial social media interactions. Ultimately, the overarching goal is

to empower stakeholders with actionable insights derived from the complex interplay of language, sentiment, and finance in the digital age.

1.6 Objective Of Research

The primary objective of this research is to develop a robust sentiment analysis model tailored specifically for comprehending and analyzing Hinglish financial social media text. To achieve this overarching goal, the following specific objectives have been delineated:

1. **Understanding Code-Mixing Complexity**: The first objective is to investigate and understand the intricacies of code-mixing phenomena prevalent in Hinglish social media conversations. This involves analyzing the patterns of language blending, identifying common linguistic structures, and discerning the impact of code-mixing on sentiment expression.

2. **Model Development**: The central focus of this research is to develop a novel sentiment analysis model capable of effectively processing Hinglish text. Leveraging deep learning techniques such as LSTM and Siamese architecture, the model will be trained to accurately identify sentiment cues amidst the linguistic diversity and code-mixing complexities inherent in financial social media interactions.

3. **Handling Domain-Specific Terminology**: Another objective is to address the challenge of domain-specific terminology prevalent in financial discourse. The sentiment analysis model will be equipped with mechanisms to recognize and interpret specialized vocabulary and jargon commonly used in financial discussions on social media platforms.

4. **Managing Context-Specific Ambiguity**: This research aims to devise strategies for managing context-specific ambiguity, a common obstacle in sentiment analysis tasks. By incorporating contextual information and domain knowledge into the model, it seeks to enhance the accuracy and reliability of sentiment predictions, particularly in scenarios where linguistic nuances may lead to ambiguous interpretations.

5. **Validation and Evaluation**: Finally, the objective includes validating and evaluating the performance of the proposed sentiment analysis model. This entails conducting comprehensive experiments using real-world Hinglish financial social media data, comparing the model's performance against existing approaches, and as-

sessing its efficacy in generating robust sentiment predictions.

By addressing these specific objectives, this research endeavors to advance the state-of-the-art in sentiment analysis methodologies, particularly in the context of multilingual financial social media interactions. Ultimately, the aim is to equip stakeholders with a powerful analytical tool capable of extracting valuable insights from the vast troves of Hinglish social media data, thereby facilitating informed decision-making and strategic planning in the financial domain.

1.7 Contribution

This thesis constitutes a pioneering effort in advancing the field of sentiment analysis within the domain of financial social media interactions, with a particular emphasis on the intricacies of code-mixed text analysis. The contributions of this research endeavor are multifaceted and promise to significantly augment our understanding and application of sentiment analysis techniques in the context of finance and fintech. The principal contributions are outlined below:

1. **Innovative Methodology Development:** A groundbreaking methodology has been conceived and executed, harnessing the power of Siamese network architecture tailored explicitly for sentiment analysis in code-mixed financial social media interactions. This innovative approach has been meticulously crafted to address the unique challenges posed by linguistic variations and nuanced sentiment expressions prevalent in financial discourse on social media platforms. By providing a robust analytical framework, this methodology equips researchers and practitioners with the tools necessary to delve deeper into the realm of financial sentiment analysis.

2. **Comprehensive Dataset Compilation:** A comprehensive dataset has been meticulously compiled, comprising code-mixed text sourced from a diverse array of financial literature, fintech forums, and social media platforms specializing in financial discussions. This curated dataset stands as a testament to the meticulous attention to detail exercised throughout the research process and serves as a valuable repository for training and evaluating sentiment analysis models tailored specifically for financial social media interactions. The availability of

such a dataset is poised to catalyze further research and experimentation in the domain of financial sentiment analysis.

3. **Insightful Analysis of Financial Sentiments:** Through rigorous analysis of sentiment in code-mixed financial text, profound insights have been unearthed regarding investor sentiments, market perceptions, and emerging financial trends prevalent in social media discourse. These insights not only enrich our understanding of the dynamics at play within financial markets but also hold significant implications for refining stock price prediction models, optimizing investment strategies, and augmenting decision-making processes in the realm of finance and fintech.

Chapter 2

Literature Reviews

2.1 Overview of Sentiment Analysis

This paper [1] delves into the intricate relationship between sentiment in financial news and stock market movements. It underscores the influence of subjective attitudes and opinions expressed in financial news on investor decisions and market volatility. Conversely, it acknowledges how market volatility can affect investor sentiment, creating a feedback loop. The paper identifies challenges in sentiment analysis due to the semi-structured nature of financial news and the variability in emotional expressions. It emphasizes the need for advanced data mining techniques to extract sentiment from vast datasets accurately.

Furthermore, the paper outlines the research objectives to enhance sentiment classification methods using data mining technologies and classification algorithms. It seeks to quantify the impact of financial news on stock market movements, potentially enabling market prediction and deeper insights into news influence. Additionally, it suggests practical implications such as guiding investors towards more informed decisions and aiding regulators in monitoring public opinion effectively for improved decision-making. Overall, the paper provides a comprehensive overview of existing research and lays out the potential contributions of its research in understanding and utilizing sentiment analysis in financial markets.

This paper [2] discusses the increasing importance of social media, particularly Twitter, as a platform for expressing opinions publicly. It highlights the vast amount of data generated on Twitter and the opportunities it provides for sentiment analysis. With millions of users and billions of tweets daily, Twitter has become a valuable

resource for researchers to analyze sentiments and make predictions about various topics, including politics, entertainment, and the stock market.

The study outlined in this paper [3] utilizes Twitter data to forecast stock prices within the American stock markets. It underscores the growing significance of social media as a font of valuable data and puts forth a technique to correlate Twitter sentiments with stock market fluctuations. The paper's objective is to categorize tweets into sentiment groups and discern correlations between public sentiments and stock prices through the application of sentiment analysis and data mining algorithms.Through experiments involving 30 representative companies and millions of Twitter records, the paper demonstrates a high prediction accuracy of 81.45%. It concludes with a summary of the proposed methodology and experimental results, emphasizing the potential of using social media data for stock market prediction.

This paper [4] explores the wealth of data generated by human interactions on social media platforms, particularly Twitter, which boasts millions of users worldwide. It highlights Twitter's suitability for sentiment analysis due to its diverse user base and the varied nature of the data produced. The paper outlines a systematic approach to compiling and analyzing opinions expressed on Twitter using the Twitter API, enabling the extraction of valuable insights into public sentiment. Through sentiment analysis and data mining techniques, the paper aims to create accurate representations of sentiment trends over time and across different locations. Additionally, it proposes using transfer learning, particularly with pre-trained BERT models like XLNET, to enhance sentiment analysis accuracy. Ultimately, the paper underscores the potential of social media data, particularly from platforms like Twitter, in informing business strategies, understanding public perception, and improving customer engagement.

2.2 Deep Learning Techniques for Sentiment Mining

This paper[5] addresses the ongoing challenges in sentiment analysis, particularly in accurately classifying sentiments across various domains. While existing approaches often focus on binary sentiment classification (positive or negative), this study introduces a novel Unsupervised Combined Sentiment-Topic (CST) model incorporating

topic detection and sentiment analysis. By considering the interplay between sentiment polarities and underlying topics in the text, the CST model aims to provide a more nuanced understanding of sentiment. Experiments conducted on multi-domain sentiment datasets demonstrate the effectiveness of the CST model in detecting positive, negative, and neutral sentiments across different domains. The paper concludes with discussions on experimental results and outlines future research directions in sentiment analysis.

The paper referenced [6] delves into the importance of sentiment analysis, especially within the financial domain, and underscores the rising interest in leveraging machine learning and deep learning methodologies to scrutinize sentiment in financial news. It accentuates the influence of sentiment analysis on investment strategies and market dynamics, referencing prior studies that have explored the correlation between sentiment and various financial indicators like trading volume, stock prices, and potential losses.

Moreover, the paper introduces a sentiment analysis approach based on a Deep Learning model employing Long Short-Term Memory (LSTM) networks to gauge sentiment towards financial news articles. Key contributions entail the formulation of a sentiment analysis model utilizing LSTMs and the assessment of performance metrics such as precision, accuracy, f1-score, and recall. The paper delineates the proposed network architecture, presents findings and discussions, and concludes by delineating potential avenues for future research endeavors.

This paper [7] discusses classification algorithms based on machine learning and deep learning techniques for sentiment analysis. It begins by outlining common supervised machine learning algorithms such as logistic regression, support vector machines, decision trees, nearest neighbour algorithms, and the XGBoost algorithm. It then delves into deep learning methods, focusing on Multilayer Perceptron (MLP) and Recurrent Neural Networks (RNN). The paper highlights the strengths and limitations of each approach and discusses their applications in sentiment classification. Additionally, it provides schematic diagrams illustrating the structure and forward propagation process of MLP and RNN. Overall, the paper aims to provide insights into the different algorithms used for sentiment analysis and their underlying mechanisms, serving as a guide for researchers and practitioners in the field.

This paper [8]explores multi-class sentiment analysis using deep learning techniques on Amazon product review data. It outlines the methodology, starting with

dataset description and preprocessing steps such as one-hot encoding, tokenization, and word embedding using Word2Vec. The paper then discusses the implementation of recurrent neural networks (RNN) and long short-term memory (LSTM) models, highlighting their architecture and advantages over traditional RNNs. Additionally, it introduces the C-LSTM model, which combines convolutional neural networks (CNN) and LSTM for text classification. The proposed methodology aims to evaluate the performance of different deep learning algorithms for sentiment analysis, providing insights into their effectiveness in processing sentiment-rich data from social media platforms like Amazon reviews.

This paper [9] addresses the increasing need for efficient sentiment analysis systems in the context of vast amounts of text data generated online, particularly focusing on product reviews. It highlights the limitations of traditional methods in handling large datasets. It emphasizes the importance of language models, particularly recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, in processing text data effectively. The paper discusses the architecture and advantages of LSTM networks over conventional RNNs, particularly in capturing long-term dependencies in text sequences. It also describes the sentiment analysis process using LSTM, including data preprocessing, training, and evaluation. Through experiments on various datasets, including English and Chinese text, the paper demonstrates the effectiveness of LSTM-based models in sentiment analysis tasks, providing insights into their performance compared to traditional RNNs.

2.3 Model Architecture - Siamese Architecture

This paper [10] introduces a model architecture based on Siamese recurrent neural networks for learning text similarity. The model is trained with a contrastive loss function and multiple embedding spaces, capturing invariances related to spelling variations, synonym replacements, and extraneous words. The study highlights the importance of incorporating prior knowledge into the system to enhance performance. Future directions include exploring additional network architectures, such as incorporating convolutional layers and investigating triplet loss functions, as well as evaluating the model on standard textual similarity and semantic entailment datasets. Further improvements could involve refining negative sample selection strategies, exploring new sources of variation, and leveraging hierarchical structures in taxonomies

for more sophisticated transfer learning.

This paper [11] introduces a Siamese LSTM-based modelling framework designed to improve document representation for text categorization tasks. By employing two Siamese LSTM sub-networks, the framework measures the semantic distance between pairs of text documents to generate representations that capture their semantic relatedness. Empirical results demonstrate promising performance compared to existing methods. Looking ahead, the authors intend to explore advanced neural network architectures and potentially apply the Siamese LSTM or CNN networks to extractive speech summarization tasks.

This paper [12] introduces MultiSiam, a modified Siamese neural network designed to handle multiple inputs simultaneously for duplicate text detection across social media platforms. Traditional Siamese models only compare pairs of inputs, limiting their effectiveness when dealing with multiple duplicates of a post. MultiSiam addresses this limitation by accommodating multiple inputs, thereby improving the efficiency of duplicate detection. The authors then propose SMCD (Social Media Classification and Duplication Model), which leverages MultiSiam to categorize social media posts based on their content and identify duplicate posts across different social media platforms. This approach streamlines the process by eliminating redundant calculations and improving the application pipeline to optimize users' social media feeds.

The paper referenced [11] investigates the utilization of a Siamese LSTM network for text categorization, with the objective of enhancing the accuracy of document representation. By harnessing the LSTM network's capability to effectively capture long-range dependencies among words, the proposed model employs pairwise learning to generate distributed document representations that encapsulate semantic distances between document pairs. Through training on pairs of documents, the Siamese LSTM network adeptly learns to generate document representations that accentuate semantic relatedness.

Experimental outcomes on standard text categorization benchmarks reveal commendable performance compared to prevailing methodologies, underscoring the efficacy of the proposed approach in acquiring more nuanced semantic representations for text categorization tasks.

The paper [13] explores various experiments using machine learning classifiers for classifying Hinglish code-mixed data. It employs ensemble voting and Tf-Idf vec-

torization techniques with different n-gram frequencies. The findings suggest that Logistic Regression exhibited superior performance across most experiments, outperforming Random Forest, SVM, Multinomial Naïve Bayes, KNN, Decision Tree, and Gaussian Naïve Bayes. The Tf-Idf vectorizer consistently yielded the most favorable outcomes, especially when employing Unigrams or Uni-Bi-Trigrams with minimum n-gram occurrence frequencies of 2 or 3. However, using only Trigrams yielded the worst results. Additionally, the normalization of Hindi words improved classifier performance, while converting emoticons into keywords decreased performance, possibly due to the nuanced nature of emoticons in conveying sentiment polarity. The study provides insights into effective techniques for classifying Hinglish code-mixed data and highlights the importance of preprocessing steps in enhancing classifier performance.

The paper [14] introduces SACMT (Sentiment Analysis of Code-Mixed Text), a novel approach for classifying sentiment in code-mixed sentences, which are common in multilingual societies on social media platforms. SACMT utilizes paired Bidirectional Long Short-Term Memory Recurrent Neural Networks (BiLSTM RNN) with shared parameters alongside a contrastive energy function to align code-mixed and standard language sentences into a unified sentiment space. Through the minimization of the energy function, SACMT effectively learns both shared model parameters and a similarity metric, enabling it to classify sentiments accurately. Additionally, a clustering-based preprocessing method addresses the challenges of transliteration variations in phonetic languages. Experimental results demonstrate SACMT's superior performance over existing approaches, highlighting its effectiveness in sentiment analysis for code-mixed text.

Chapter 3

Background

3.1 Machine Learning

Machine learning is a pivotal field in artificial intelligence, empowering computers to acquire knowledge and enhance their performance without explicit programming by humans. It encompasses a diverse set of techniques aimed at extracting patterns and insights from data. Machine learning is categorized into three main types: supervised, unsupervised, and reinforcement.

- **Supervised Learning:** Supervised learning algorithms are trained using labelled data, where each data point is associated with a corresponding label. During training, the model learns to predict outcomes based on input-output pairs provided in the labelled dataset. When labelled data is available, supervised learning is highly effective for sentiment classification tasks.

  Pros:

  - Effective for sentiment classification when labelled data is available.
  - Capable of handling both binary and multi-class sentiment analysis tasks.
  - Well-established algorithms with clear principles.

  Cons:

  - Acquiring a substantial amount of labeled data is often resource-intensive, involving significant costs and time investment.
  - Prone to overfitting if not properly regularized.

- **Unsupervised Learning:** Unsupervised learning algorithms are trained on unlabelled data, where the model discovers hidden patterns and structures. Unlike supervised learning, no explicit labels are provided during training. Common tasks in unsupervised learning include clustering, where data points are grouped based on similarities.

  Pros:

  - Useful for discovering sentiment patterns in unlabelled data.
  - Can identify clusters of similar sentiments without needing labelled data.

  Cons:

  - May produce less accurate results compared to supervised methods due to the absence of labelled data.
  - Interpretability can be challenging as the model learns patterns without human supervision.

- **Reinforcement Learning:** Reinforcement Learning (RL) is a machine learning approach wherein an agent learns to navigate an environment by taking actions and receiving feedback in the form of rewards or penalties. The objective of reinforcement learning is to determine a policy or action strategy that maximizes the agent's total reward across successive interactions. RL algorithms iteratively adjust the agent's actions based on the observed rewards, aiming to optimize its long-term performance.

  Pros:

  - Can optimize decision-making strategies for sentiment-based actions.
  - Learns through trial and error, which can lead to adaptive behaviour in dynamic environments.

  Cons:

  - Complex to implement and may require extensive computational resources and training time.
  - May be less suitable for sentiment analysis than other methods due to its focus on sequential decision-making.

These categories denote distinct methodologies in machine learning, each tailored to address various problem types and data structures. Supervised learning revolves around predicting outcomes using labelled data examples, unsupervised learning delves into uncovering hidden patterns within unlabelled datasets, while reinforcement learning empowers agents to refine decision-making strategies through iterative trial and error. Each approach offers unique strengths and weaknesses, underscoring the importance of selecting the most suitable technique based on the particular demands and attributes of the sentiment analysis task under consideration.

3.2 Traditional sentiment classification techniques

Traditional sentiment analysis techniques encompass various machine learning algorithms and methods for classifying text data into sentiment categories. Here are some commonly used techniques:

1. **Naive Bayes (NB)**: A Bayes-based probabilistic classifier that assumes feature independence. NB determines a document's likelihood of incurring into a specific sentiment class by looking at the characteristics that appear in the content.

2. **Support Vector Machine (SVM)**: SVM is a supervised learning algorithm that separates data points into different classes by finding the hyperplane that maximizes the margin between classes. SVM classifiers learn to differentiate between positive and negative sentiment by mapping text features to a high-dimensional space.

3. **Logistic Regression (LR)**: A statistical model used for binary classification tasks. LR models use a logistic function to estimate the probability of a text belonging to a particular sentiment class. They learn the relationship between input features (words or n-grams) and the likelihood of a document being positive or negative.

4. **Decision Trees**: DT is a hierarchical structure that recursively partitions the feature space based on the values of input features. Decision tree classifiers make decisions by following a path from the root node to the leaf node, where each node represents a decision based on a feature value.

5. **Random Forests**: RF is an ensemble learning method that constructs multiple decision trees, training each tree on a random subset of the training data and features. The final prediction is then determined through voting or averaging the predictions of individual trees, resulting in a robust and accurate model.

6. **K-Nearest Neighbors (KNN)**: It is a non-parametric classification algorithm that k nearest neighbors of a new point in the feature space, to decide which class it belongs to. KNN calculates the similarity between the input text and training examples using distance metrics such as Euclidean distance or cosine similarity.

7. **Ensemble Methods**: Techniques that combine multiple base classifiers to improve prediction accuracy and robustness. Examples include bagging, boosting, and stacking. Ensemble methods leverage the diversity of base classifiers to handle different aspects of sentiment and reduce overfitting.

These techniques offer a variety of approaches to sentiment analysis, each with its strengths and weaknesses. The choice of technique depends on factors such as the data's nature, the dataset's size, and the computational resources available.

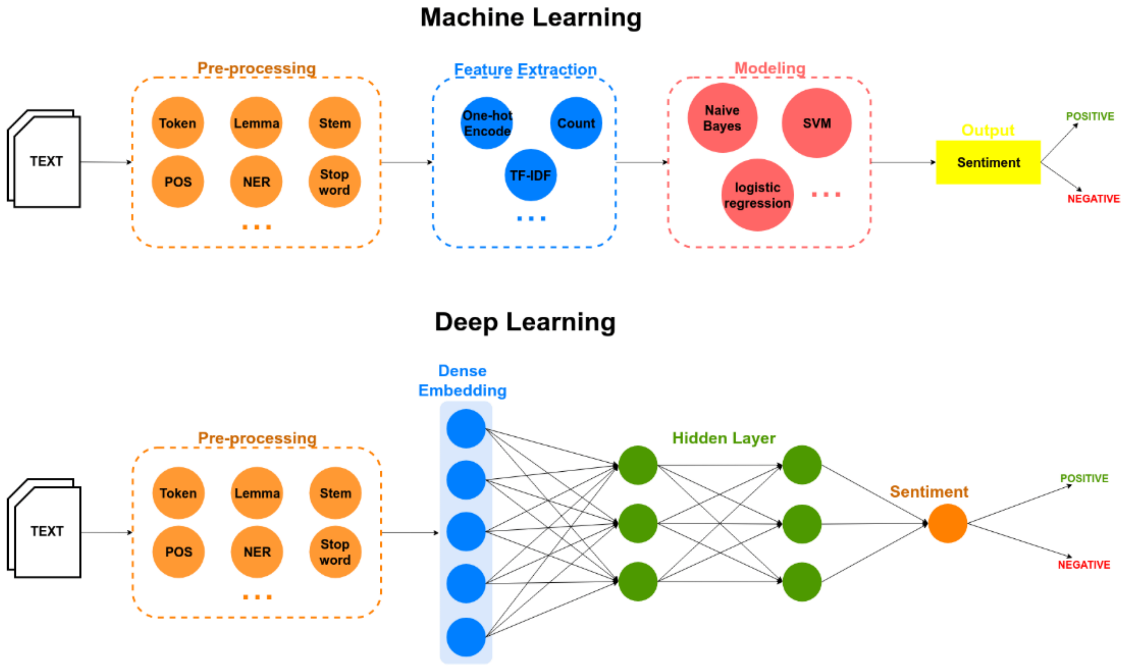

Figure 3.1: Machine Learning and Deep Learning

3.3 Deep Learning

Deep learning revolutionizes neural networks by introducing multiple layers into their hidden layers, empowering them to grasp intricate patterns and correlations within data. In contrast to conventional machine learning approaches, which require manual feature engineering or feature selection, deep learning models autonomously learn and extract relevant information directly from raw data, resulting in improved accuracy and performance. Furthermore, hyperparameters of classifier models in deep learning are often automatically adjusted, simplifying the model optimization process.

Figures 3.1 and 3.2 depict a comparison between traditional machine learning techniques like Support Vector Machine (SVM), Bayesian networks, decision trees, and deep learning methods for sentiment polarity categorization. Deep learning has emerged as the preferred method for addressing complex challenges across domains such as image recognition, voice recognition, and natural language processing (NLP), consistently delivering cutting-edge solutions. In the subsequent sections, we will explore various deep-learning methodologies utilized in sentiment analysis and related tasks.

3.3.1 Deep Neural Network

Deep neural networks represent a new generation of machine learning algorithms inspired by the structure and function of the human brain. Their distinctive characteristic lies in their ability to automatically extract relevant features from raw data. A deep learning model can be expressed as a mathematical function $f : X \rightarrow Y$, where X represents the input data and Y represents the output.

Deep learning extends the concept of Artificial Neural Networks (ANN) by incorporating multiple hidden layers to model complex datasets [**?**]. Illustrated in Figure 3.2, a typical deep neural network consists of three primary layers:

- **Input Layer**: Neurons in this layer receive input data from the variable X.
- **Hidden Layers**: These layers consist of neurons that receive signals from the preceding input layer. Each hidden layer autonomously learns its own set of features, with deeper layers capturing increasingly abstract representations.
- **Output Layer**: Comprised of neurons that receive input from the hidden layers and produce the final output value.

Deep neural networks have demonstrated exceptional performance across various domains, owing to their capability to learn hierarchical representations from data automatically.

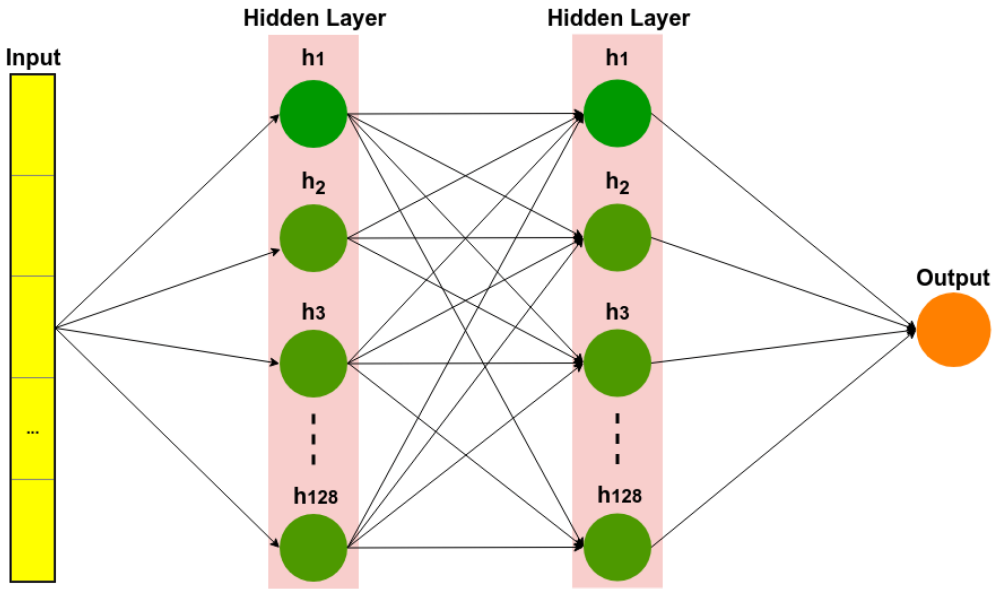

Figure 3.2: Deep Neural Network

3.3.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a category of deep learning architectures comprising multiple layers of convolutions accompanied by nonlinear activation functions like ReLU or tanh. Unlike traditional feedforward neural networks, where every input neuron is connected to every output neuron in the subsequent layer (fully connected or affine layer), CNNs compute outputs via convolutions over the input layer. This arrangement fosters local connections, wherein each input region is linked to a neuron in the output layer.

In CNNs, individual layers employ numerous filters, often numbering in the hundreds or thousands, and amalgamate the resulting feature maps. Throughout the training process, a CNN autonomously learns the filter values based on the intended task. For instance, in image classification tasks, a CNN might learn to discern edges from raw pixels in its initial layer, then utilize these detected edges to identify basic shapes in subsequent layers, and finally leverage these basic shapes to recognize

higher-level features, such as facial contours, in deeper layers. The final layer of a CNN typically functions as a classifier, utilizing these high-level features to make predictions.

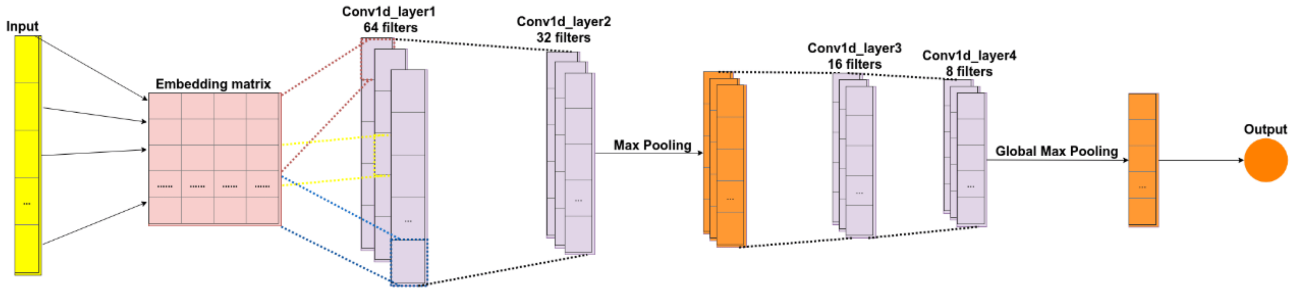

Figure 3.3: Illustration of a Convolutional Neural Network (CNN) architecture for sentiment analysis.

In Natural Language Processing (NLP), tasks typically involve input matrices consisting of phrases or texts, where each row corresponds to a token (typically a word or character). These tokens are commonly encoded using word embeddings like word2vec or GloVe, or as one-hot vectors indexing the word within a dictionary.

- **Pros:**
  - Effective in capturing local patterns and dependencies within textual data, allowing them to identify relevant features for sentiment classification.
  - Automatically acquire distinctive features from raw text, obviating the necessity for manual feature engineering.
  - Architectures can be designed with multiple convolutional and pooling layers, enabling the model to learn hierarchical representations of text at different levels of abstraction.
  - Computationally efficient, especially when processing large volumes of text data, making them scalable for real-world sentiment analysis applications.
- **Pros:**
  - Effective in capturing local patterns and dependencies within textual data, allowing them to identify relevant features for sentiment classification.
  - Automatically acquire distinguishing features from raw text, removing the requirement for manual feature engineering.

- Architectures can be designed with multiple convolutional and pooling layers, enabling the model to learn hierarchical representations of text at different levels of abstraction.
- Computationally efficient, especially when processing large volumes of text data, making them scalable for real-world sentiment analysis applications.

3.3.3 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are an extension of feedforward neural networks equipped with an internal memory mechanism [**?**]. Unlike feedforward networks, RNNs exhibit a recurrent nature, performing the same computation for each input while considering the previous calculations. The output generated at each time step is fed back into the network, allowing it to maintain memory and capture sequential dependencies in the data. However, conventional RNNs encounter the issue of the vanishing gradient problem, which restricts their capacity to grasp long-term dependencies.

Long Short-Term Memory Networks (LSTMs) were introduced to address this issue as an advanced form of RNNs [**?**]. LSTMs incorporate specialized memory cells that allow for the persistence of information over time, mitigating the vanishing gradient problem and enabling the model to remember long-term dependencies. In the preprocessing step, input data is reformatted for the embedding matrix, similar to the process described for CNNs. Subsequently, the input is passed through an LSTM layer with 200 cells. Finally, a fully connected layer with 128 text categorization units is utilized, with the output vector being reduced to a single output representing the sentiment class (positive or negative) using the sigmoid activation function.

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = \text{softmax}(W_{hy}h_t + b_y)$$

In this context, x_t denotes the input at time step t, h_t signifies the hidden state at time step t, and y_t represents the output at time step t. The parameters W_{xh}, W_{hh}, W_{hy} denote weight matrices, while b_h, b_y stand for bias vectors. The function σ typically refers to a nonlinear activation function such as the sigmoid or hyperbolic

tangent function, and softmax indicates the softmax function employed for multi-class classification tasks.

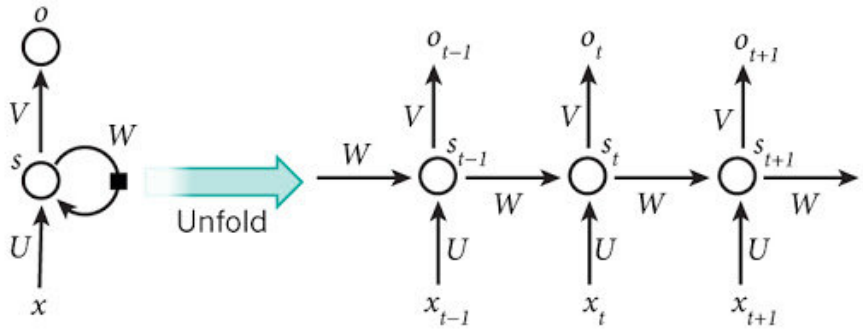

Figure 3.4: Illustration of RNN Architecture

- **Pros:**
  - Effective in capturing sequential dependencies and long-range contextual information in text data, which is crucial for sentiment analysis tasks.
  - Suitable for handling variable-length input sequences, making them adaptable to different text lengths and structures.
  - Can model complex relationships between words and phrases over time, allowing for a deeper understanding of sentiment nuances.
  - Well-suited for sequential decision-making tasks, such as sentiment analysis in dialogue systems or time-series sentiment analysis.
- **Cons:**
  - Complex to implement and may require extensive computational resources and training time, especially for deep architectures or large datasets.
  - Vulnerable to the vanishing or exploding gradient problem, which can hinder the learning of long-term dependencies and degrade performance.
  - Prone to overfitting, especially when dealing with small datasets or noisy text inputs, leading to reduced generalization performance.
  - Interpretability can be challenging, as RNNs lack transparency in their decision-making process, making it difficult to understand how specific sentiments are inferred.

3.3.4 Long Short-Term Memory Networks (LSTMs)

A specific kind of RNN called a LSTM was created to solve the vanishing gradient issue and identify long-range dependencies in sequential data. LSTMs, in contrast to conventional RNNs, have memory cells and gating mechanisms that allow them to remember or forget information over time selectively.

The core components of an LSTM cell include the input gate (i_t), the forget gate (f_t), the output gate (o_t), and the cell state (c_t). These gates control the flow of information within the LSTM cell, allowing it to regulate the input, forget unnecessary information, update the cell state, and produce the output at each time step.

The following equations govern the computations within an LSTM cell:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Here, x_t represents the input at time step t, h_{t-1} denotes the hidden state from the previous time step, and W and b are the weight matrices and bias vectors, respectively, for the input, hidden, and cell state connections. Additionally, σ represents the sigmoid activation function, and $\odot$ denotes element-wise multiplication. Figure 3.5 illustrates an LSTM cell's architecture and its information flow.

- **Pros:**
  - Effective in capturing long-term dependencies and preserving contextual information over extended sequences, making them well-suited for sentiment analysis tasks.
  - Mitigate the vanishing gradient problem, allowing them to learn and retain information over many time steps, leading to improved performance in modelling sequential data.

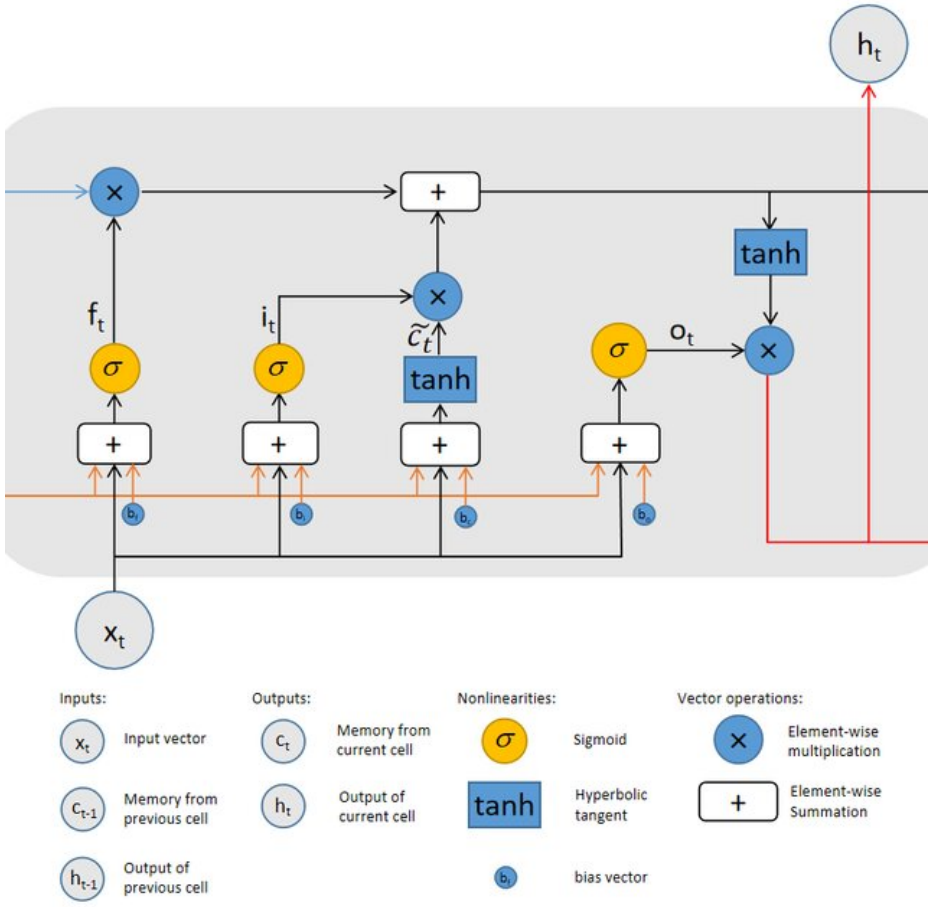

Figure 3.5: Illustration of an LSTM cell architecture.

  - Can handle variable-length input sequences and adaptively adjust their memory cells, enabling them to process text of varying lengths and structures.
  - Versatile architecture suitable for various NLP tasks beyond sentiment analysis, such as machine translation, text generation, and named entity recognition.

- **Cons:**
  - More computationally intensive than simpler RNN architectures, requiring additional training time and computational resources.
  - Vulnerable to overfitting, especially when dealing with small datasets or noisy text inputs, necessitating regularization techniques such as dropout

or weight decay.

- May have difficulty interpreting the internal states and memory dynamics of the LSTM cells, limiting the model's transparency and interpretability.
- Hyperparameter tuning for LSTM architectures can be challenging, requiring careful optimization of parameters such as the number of layers, hidden units, and dropout rates.

3.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art natural language processing (NLP) framework [**?**]. Unlike traditional word embedding methods that rely on contextual information from surrounding words, BERT leverages the bidirectional nature of Transformers. Each output element is connected to every input element in this deep learning architecture. This allows BERT to capture complex relationships between words in both directions, significantly enhancing its ability to understand context.

BERT consists of an encoder that processes input text by transforming tokens into vectors and passing them through multiple layers of neural networks. The model is trained on large-scale text corpora using unsupervised learning techniques, allowing it to learn rich language representations. BERT has demonstrated exceptional performance across a wide range of NLP tasks, surpassing previous benchmarks in seven out of eleven tasks. Alternative versions of BERT, such as Distilled BERT, GPT-3, and XLNet, have also been developed, but BERT remains the dominant choice due to its superior performance.

- **Pros:**
  - BERT can capture complex linguistic patterns and nuances present in Hinglish, a blend of Hindi and English, due to its pretraining on large-scale text corpora.
  - The bidirectional nature of BERT allows it to effectively understand context and semantics in Hinglish sentences, enabling accurate sentiment analysis even in multilingual environments.

  - BERT's contextual embeddings can adapt to the diverse vocabulary and syntax of Hinglish, providing robust representations of words and phrases in this hybrid language.
  - BERT's fine-tuning mechanism allows for efficient adaptation to sentiment analysis tasks in Hinglish, achieving high accuracy with minimal task-specific labelled data.

- **Cons:**
  - Fine-tuning BERT for sentiment analysis on Hinglish datasets may require substantial labelled data to achieve optimal performance, as BERT's pre-trained weights are primarily based on English text.
  - BERT's computational requirements for fine-tuning and inference can be demanding, particularly for large Hinglish datasets, necessitating access to powerful hardware or cloud-based computing resources.
  - BERT's model size and complexity may lead to challenges in deployment, especially in resource-constrained environments where memory and processing power are limited.
  - Despite its effectiveness, BERT's performance on sentiment analysis in Hinglish may be influenced by the availability and quality of labelled data, domain-specific vocabulary, and linguistic variations across different regions and dialects.

3.5 Performance Measures

The classifier's performance is assessed using metrics different from the model's performance. The optimal metrics, which consist of text data and multi-class classification, are selected as the data used for the thesis. The accuracy, recall, F-score, and area under the curve are prominent measures in multi-class classification.

3.5.1 Accuracy

Classification accuracy is the total number of correct predictions divided by the classifier's total number of predictions. A 'true positive' results when the model accurately

predicts the positive class. Likewise, a 'true negative' results in the model accurately predicting the negative class. A 'false positive' result from the model mistakenly predicts the negative class as positive. A 'false negative' result from the model mistakenly predicts the positive class as a negative class. For a binary classification problem, the accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

3.5.2 Precision

Precision is another statistic used to measure the performance of a classifier. It is determined by dividing the number of positive class predictions, i.e., true positives that belong to the positive class, by the total number of true positives.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3.5.3 Recall

Recall is calculated as true positive divided by the true positives and false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.5.4 F1 Score

The F1 score is the harmonic mean of accuracy and recall, accounting for both measurements using the following equation:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

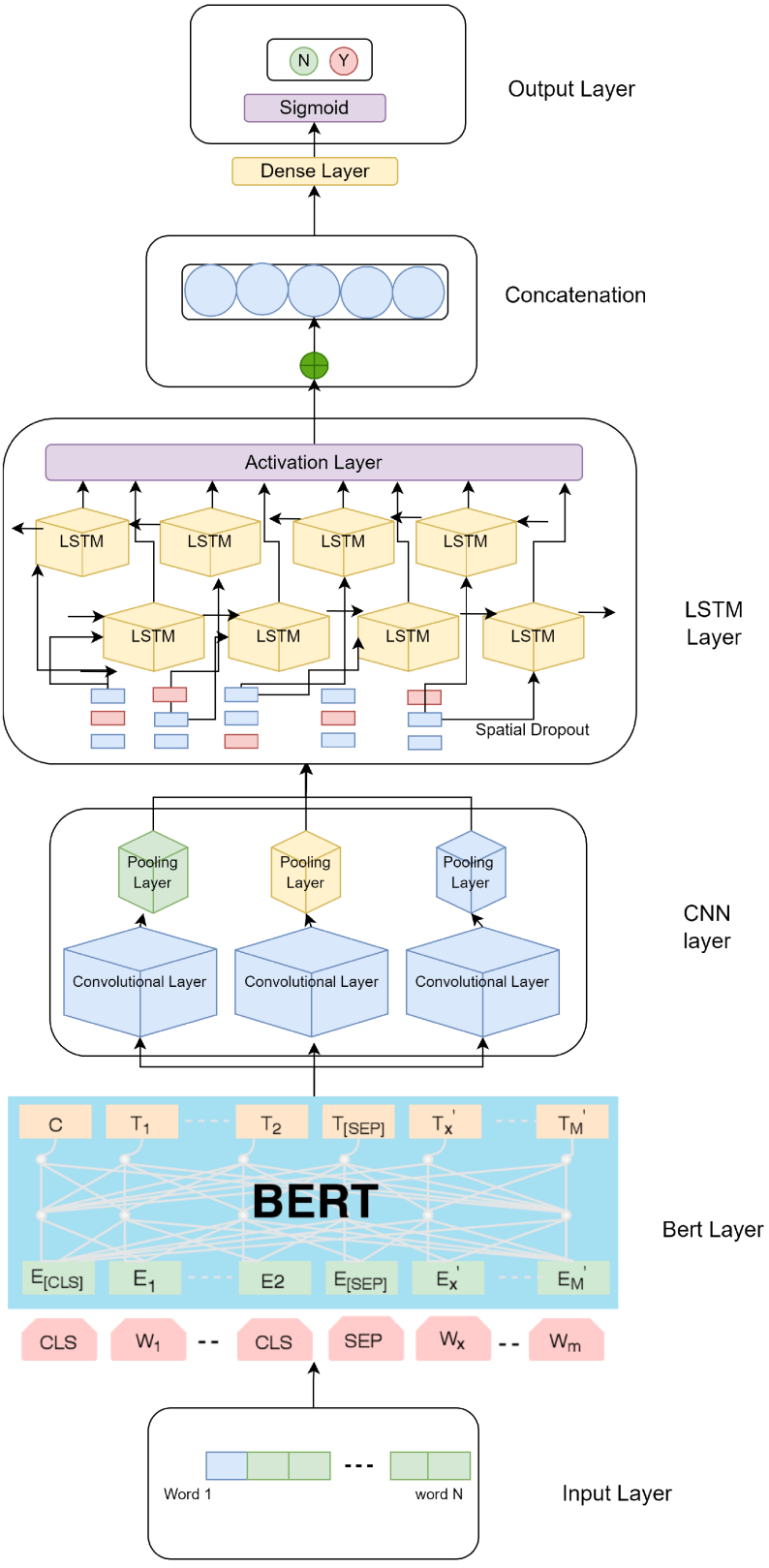

Figure 3.6: BERT Architecture

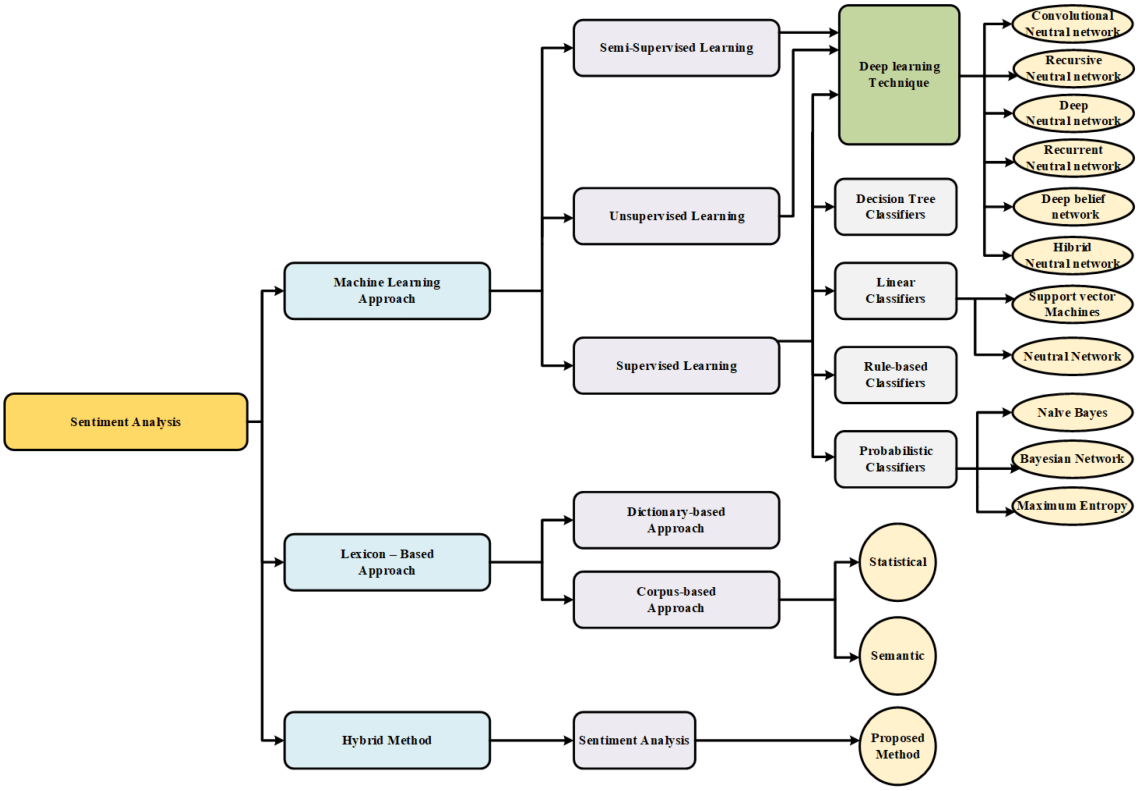

Figure 3.7: Categorisation of Sentiment Analysis Techniques

Chapter 4

Methodologies for Model Generation

4.1 Word embedding

One of the challenges that we face in text processing is making computers understand what a word means. For computers, characters are just symbols with no meaning attached to them. But they're much more to us. We understand words based on their meanings and how they relate to each other.
Now, word embedding is like giving the computer a way to translate words into numbers so that computers can understand. But these aren't just random numbers. They're carefully crafted to represent the meanings of words and how they're used together. So, feeding a piece of text into a computer converts each word into a unique set of numbers. The aim is to add context to the number representation of the word so that the computer can make sense of the text in its language.

4.1.1 Word2Vec

Word2Vec is a widespread technique in natural language processing (NLP) for creating word embeddings. It was developed by a team of researchers at Google, led by Tomas Mikolov, in 2013. The original Efficient Estimation of Word Representations in Vector Space paper introduced two architectures, the Continuous Bag of Words (CBOW) and Skip-gram models. Word2Vec learns from these predictions and adjusts words' numerical representations (vectors) so that similar words have similar vectors.

This way, words with similar meanings or usage end up closer in the vector space.

CBOW: Continuous Bag of Words (CBOW) is a fundamental technique in natural language processing (NLP) that aims to predict the centre word of a sentence based on the context words surrounding it. By analysing the words near the target word, CBOW effectively captures the text's semantic meaning and syntactic structure. Imagine solving a puzzle where each word represents a piece, and CBOW acts as the solver, strategically analysing the clues provided by the neighbouring words to determine the missing piece—the centre word. This approach allows CBOW to grasp the intricate nuances of language, discerning subtle variations in meaning and context. Through its ability to comprehend the contextual cues embedded within a sentence, CBOW is a powerful tool for tasks such as language modelling, sentiment analysis, and machine translation, contributing significantly to advancements in NLP research and applications.

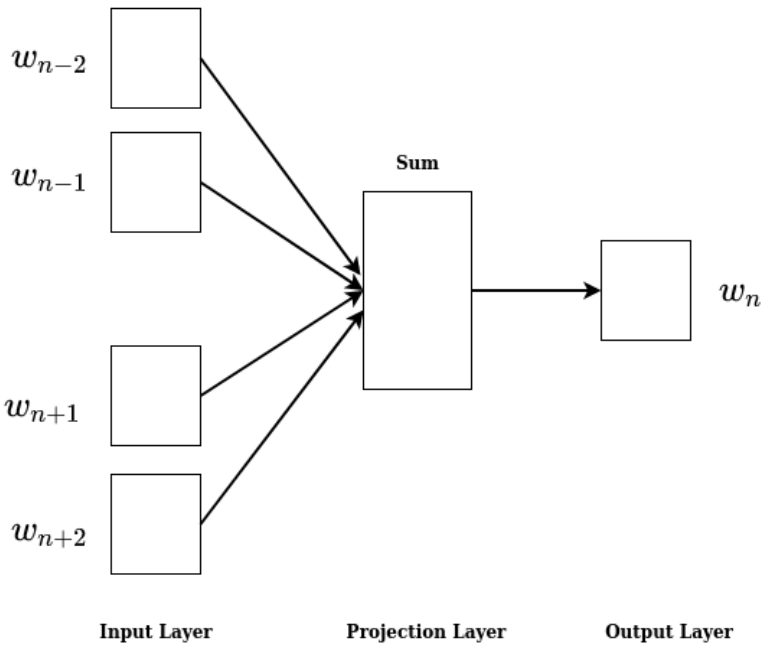

Figure 4.1: CBOW in Word2Vec

Input to the model is the concatenation of one-hot encoded vectors. The model uses two types of weight matrices, U and V.

Activation Function: Since Softmax provides the output vector as probabilities of occurrence for each word in the corpus, it is utilised as the activation function for the final layer. Lastly, the term with the highest likelihood is forecast.

$$h = \mathbf{X} \cdot \mathbf{U} \tag{4.1}$$

$$z = \mathbf{h} \cdot \mathbf{V} \tag{4.2}$$

$$\mathbf{p_i} = \boldsymbol{softmax(z)} \tag{4.3}$$

Where:

- **X** is the input vector,
- **U** and **V** are the weight vectors.

Loss Function: The loss function used for CBOW is logarithmic loss.

$$\mathrm{L} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right) \tag{4.4}$$

The weight matrices are then adjusted with the help of backpropagation. The final output of the model gives the vector of the respective word. The words with similar context words tend to occur close to each other.

Skip-gram model: Skip-gram, a popular model in natural language processing (NLP), stands in contrast to the Continuous Bag of Words (CBOW) model. While CBOW predicts the target word based on context, Skip-gram does the opposite. It uses the target word w_t to predict the words in their context, represented by w_c Skip-gram aims to learn word embeddings by maximising the probability of predicting context words given the target word. This is achieved by iteratively adjusting the model parameters to improve its predictions. By doing so, Skip-gram captures the semantic relationships between words and produces dense vector representations that can capture the meaning and context of words in a high-dimensional space. One of the advantages of Skip-gram is its ability to handle large vocabularies efficiently. Since it only needs to predict the context words given the target word, it can focus on a smaller set of training examples than CBOW, which needs to predict the target

word given its context.

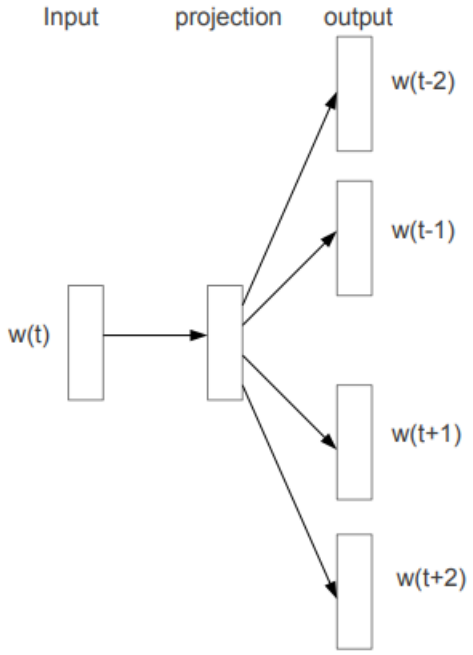

Figure 4.2: Skip-gram in Word2Vec

$$h = \mathbf{w_t} \cdot \mathbf{U} \tag{4.5}$$

$$z = \mathbf{h} \cdot \mathbf{V} \tag{4.6}$$

finally,

$$\mathbf{p_i} = \boldsymbol{softmax(z)} \tag{4.7}$$

The loss function for the Skip-gram model is typically defined using negative log-likelihood. Let $P(w_c|w_t)$ be the probability of context word w_c given target word w_t. Then, the loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\log P(w_c|w_t) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\log\frac{\exp(z_{tc})}{\sum_{j=1}^{V}\exp(z_{tj})} \tag{4.8}$$

Where:

- T is the total number of target words,

- C is the total number of context words,
- $P(w_c|w_t)$ is the conditional probability of context word w_c given target word w_t.

4.1.2 GloVe embedding

GloVe (Global Vectors for Word Representation) is another widespread word embedding technique. Developed by Stanford, GloVe is a word embedding technique that aims to capture the global co-occurrence statistics of words in a corpus to learn word representations. Unlike traditional approaches like Word2Vec, which focus on local context (e.g., window-based contexts), GloVe leverages global word-word co-occurrence statistics to build dense vector representations for words.
Modelling the link between words based on their co-occurrence probability is the central concept behind GloVe. The co-occurrence matrix X captures the frequency with which words occur together within the confines of a set window size.

GloVe Word-Word Co-occurrence Probability:

$$P_{ij} = \frac{X_{ij}}{X_i} \tag{4.9}$$

GloVe Model Equation:

$$\mathbf{w}_i^T \mathbf{w}_j + b_i + b_j = \log(X_{ij}) \tag{4.10}$$

GloVe Loss Function:

$$\mathcal{L} = \sum_{i,j=1}^{|V|} f(X_{ij}) \left(\mathbf{w}_i^T \mathbf{w}_j + b_i + b_j - \log(X_{ij})\right)^2 \tag{4.11}$$

4.1.3 FastText

FastText, an expansion of the Word2Vec model, has a distinct approach to word encoding, making it especially useful for handling languages other than English. In contrast to conventional word embedding methods, which handle words as discrete units, FastText encodes words as n-grams or bags of characters. FastText can better cap-

ture morphological and semantic information with this subword modelling approach, especially for languages like Turkish, Finnish, or Hungarian with rich morphology or agglutinative structures. FastText can handle uncommon and out-of-vocabulary words skillfully by splitting words into smaller pieces, giving meaningful representations even for terms not seen during training.
FastText is relatively language agnostic due to its subword approach, which gives it flexibility beyond linguistic diversity. This implies that the model architecture of FastText doesn't need to be significantly changed to accommodate languages with diverse scripts, morphologies, and vocabularies. This flexibility is beneficial when working with under-resourced languages or multilingual environments where labelled data and massive corpora are hard to come by. FastText's ability to capture subword information makes it well-suited for cross-lingual applications. By learning embeddings based on character n-grams, FastText can capture similarities between words in different languages, facilitating transfer learning and cross-lingual tasks such as bilingual lexicon induction or cross-lingual document classification.

Subword Embedding: With FastText, subword embedding is a method for representing words as bags of character n-grams, where n usually has a value between three and six. Prefixes, suffixes, and morphemes are examples of the smaller components that FastText divides words into rather than treating them as a single entity. This method improves FastText's ability to capture semantic and morphological information, particularly for agglutinative or richly morphological languages.

$$\mathbf{v}_w = \sum_{g \in G(w)} \mathbf{v}_g \tag{4.12}$$

Word Probability Prediction:

$$P(w|c) = \frac{e^{\mathbf{v}_w \cdot \mathbf{v}_c}}{\sum_{w' \in V} e^{\mathbf{v}_{w'} \cdot \mathbf{v}_c}} \tag{4.13}$$

FastText Loss Function:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \log P(w_i|c_i) \tag{4.14}$$

LSTM Extended Short-Term Memory Networks (LSTMs) are a specialised type of recurrent neural network (RNN) designed to address the vanishing gradient problem and capture long-range dependencies in sequential data [**?**]. Unlike traditional RNNs, LSTMs incorporate memory cells and gating mechanisms that enable them to retain or forget information over time selectively.

The core components of an LSTM cell include the input gate (i_t), the forget gate (f_t), the output gate (o_t), and the cell state (c_t). These gates control the flow of information within the LSTM cell, allowing it to regulate the input, forget unnecessary details, update the cell state, and produce the output at each time step.

The following equations govern the computations within an LSTM cell:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Here, x_t represents the input at time step t, h_{t-1} denotes the hidden state from the previous time step, and W and b are the weight matrices and bias vectors, respectively, for the input, hidden, and cell state connections. Additionally, σ represents the sigmoid activation function, and $\odot$ denotes element-wise multiplication.

Figure 3.5 illustrates an LSTM cell's architecture and its information flow.

4.2 Siamese Neural Network

A Siamese neural network is a unique network consisting of several subnetworks. The identical subnetworks share the same weights and parameters. The objective of the Siamese neural network is to learn a function that could calculate the similarity or dissimilarity between a pair of inputs leveraging on the feature values of each input. Siamese networks are frequently utilised when determining the degree of similarity between two input pairs, which is the aim of a task like similarity measurement, matching, and verification. Pairs of inputs are usually supplied to the Siamese network during training, along with labels designating whether the inputs are similar or

dissimilar. When dealing with identical inputs, the network learns to reduce the distance between feature representations and maximise the distance between dissimilar inputs. A contrastive or triplet loss function is used to accomplish this, encouraging the network to identify between similar and dissimilar pairs accurately. Siamese networks are widely used in tasks requiring similarity measurement, including image similarity, facial recognition, signature verification, and natural language processing tasks like sentence similarity and paraphrase identification. Siamese networks provide a strong tool for various similarity-based tasks by learning to collect and evaluate the tiny variations and similarities between pairs of inputs. This allows for reliable and robust performance even when there is a lack of labelled data.

4.2.1 Architecture

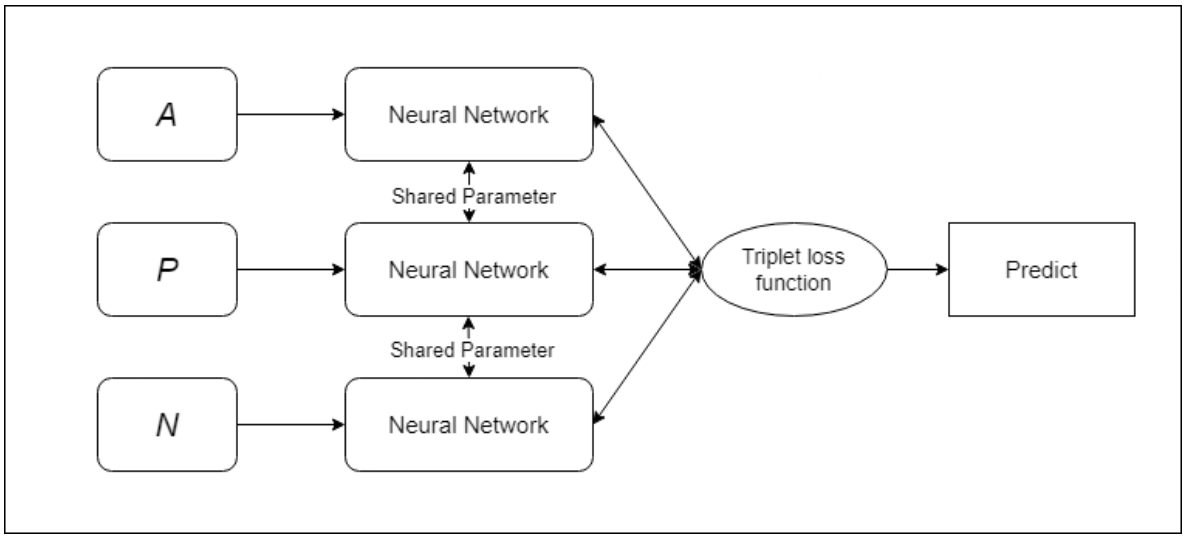

Figure 4.3: Siamese network with triplet loss function

The architecture of a Siamese network with a triplet loss function is designed to learn embeddings for input data points in a way that encourages similar data points to be closer together in the embedding space while pushing dissimilar data points farther apart. At the heart of this architecture lies the triplet loss function, which facilitates learning by enforcing a margin-based constraint on the relative distances between anchor, positive, and negative examples within each triplet.

The triplet loss function compares the distances between the anchor and positive examples with the distance between the anchor and negative examples. Let $d(A, P)$

denote the Euclidean distance between the anchor A and positive example P, and $d(A, N)$ denote the Euclidean distance between the anchor A and negative example N. The triplet loss $\mathcal{L}_{\text{triplet}}$ is defined as:

$$\mathcal{L}_{\text{triplet}} = \max\{d(A, P) - d(A, N) + \alpha, 0\} \tag{4.15}$$

where α is a margin hyperparameter that controls the minimum difference between the distances of positive and negative examples. The triplet loss aims to minimise the distance between similar examples (anchor and positive) while maximising the distance between dissimilar examples (anchor and negative).

Chapter 5

Proposed Methodology

The chapter dedicated to dataset collection, pre-processing, and technological methodologies is the foundation for this thesis. Herein lies a detailed account of the meticulous process undertaken to curate and prepare the dataset essential for the subsequent analysis. From the intricate steps involved in data collection to the sophisticated techniques employed in pre-processing, this chapter elucidates the methodology employed to ensure the integrity and relevance of the dataset. Additionally, it delves into the technological tools and frameworks leveraged to facilitate this process, shedding light on the intricacies of data handling and manipulation. This chapter lays the groundwork for this thesis's subsequent analysis and findings through a comprehensive exploration of dataset collection, pre-processing methodologies, and technological underpinnings.

5.1 Collection of data

Data collection for this study involved leveraging the capabilities of advanced natural language processing (NLP) models, specifically ChatGPT and Claude.ai, to generate realistic conversations between experienced stock traders discussing financial market trends and stock market movements. These conversations were designed to be rich in financial terminology and expressions commonly used in the industry, focusing on incorporating Hinglish language to reflect real-life interactions among traders from diverse linguistic backgrounds.

By leveraging advanced NLP models and carefully designing scenarios based on real-life incidents and market trends, the collected data provides valuable insights into

the dynamics of stock trading and financial markets. These simulated conversations offer a unique perspective on how experienced traders analyze market conditions, make investment decisions, and navigate the complexities of the financial landscape.

5.2 Data preprocessing

Data processing is like preparing a puzzle before putting it together. You start by gathering all the puzzle pieces (like your data points), ensuring they're clean and not missing any pieces (like checking for errors or missing values). Data preprocessing is getting your ingredients ready before cooking a meal. Data preprocessing is about preparing your data before feeding it into a machine-learning model, such as cleaning, chopping, and organising your ingredients before cooking. Then, you might sort them into groups based on their colours or patterns (which is like organizing your data). After that, you might resize or reshape some pieces to fit better (like normalizing or standardizing your data). Finally, you put the puzzle together, piece by piece, until you have a complete picture (which is like training your machine learning model). Data processing prepares your data for analysis or modelling, ensuring it's in the right shape and form for the task at hand.

5.2.1 Removing Stopwords

Stopwords refer to common words found in a language, such as the, is, and, of, and others. These words lack significant meaning in the context of natural language processing tasks. As a result, they are frequently eliminated from text data before analysis or modelling.

The removal of stopwords serves two primary purposes: to minimize noise and enhance the efficiency and accuracy of natural language processing tasks. By filtering out these common words, the focus shifts to more meaningful terms that convey the essence of the text and are pertinent to the specific analysis or modelling objectives. This preprocessing step enables algorithms better to discern the salient features and patterns within the text, facilitating more effective analysis and interpretation of the data.

5.2.2 Tokenization

Tokenization is akin to dissecting a piece of text into its fundamental building blocks, like pulling apart a jigsaw puzzle into its individual pieces. Each token acts as a puzzle piece, carrying its own unique significance within the context of the text. We can produce a more organized representation of the text that computers can understand and analyze more efficiently by segmenting the text into these tokens. It's similar to giving a computer a set of Lego bricks rather than a disorganized assortment of parts; this allows the computer to construct complex models, such chatbots or language processing algorithms, more precisely and clearly. Furthermore, depending on the particular requirements of the task at hand, tokenization is more than just separating words; it's about capturing the substance of language, whether it's identifying words, phrases, or even special characters.

We have utilized the tokenizer offered by keras.preprocessing.text package. We have successfully processed and analyzed the text by utilizing Keras' strong tokenization capabilities, ensuring precision and efficiency in our tasks.

5.2.3 Lemmatization

Lemmatization is a key technique in natural language processing approaches, providing an advanced way to normalize and reconcile lexical variations in textual data. Lemmatization helps to provide a deeper grasp of the semantic linkages within the text by identifying the morphological subtleties of words and distilling them down to their lemma. This procedure improves the accuracy and consistency of computational studies and encourages more complex interpretations of language phenomena, which enriches academic research. Furthermore, lemmatization is essential for reducing the effects of vocabulary inconsistencies in fields like computational linguistics, information retrieval, and sentiment analysis. This helps to maintain the stability and effectiveness of computational models and algorithms used in research.

WordNetLemmatizer is used to achieve the goal of lemmatization. It is a powerful tool within the NLTK (Natural Language Toolkit) library and stands as a cornerstone in achieving lemmatization goals. By leveraging the expansive lexical database of WordNet, this lemmatizer excels in mapping words to their corresponding lemma forms with exceptional accuracy and efficiency.

5.3 Data Preparation

The proposed architecture uses a Siamese neural network with a triplet loss function. A triplet consists of an anchor sample A_i, a positive sample P_i, i.e. a sample from the dataset that has same label as anchor, and a negative sample N_i, i.e. a sample from the dataset that is of different class than the anchor. The samples are chosen for triplets from the dataset with replacement. Each sample from the triplet undergoes feature extraction through these subnetworks, producing embedding vectors that capture the essential characteristics of the input data. The network is trained to minimize the distance between the embeddings of A_i and P_i samples while maximizing the distance between A_i and N_i samples, as dictated by the triplet loss function.

5.4 Word Embedding

Word embedding is like giving the computer a way to translate words into numbers so that computers can understand. But these aren't just random numbers. They're carefully crafted to represent the meanings of words and how they're used together. So, feeding a piece of text into a computer converts each word into a unique set of numbers. The aim is to add context to the number representation of the word so that the computer can make sense of the text in its own language.
A matrix for embeddings created by optimizing Word2Vec vectors that have already been acquired. Using our particular dataset, we were able to further train the Word2Vec embeddings as well as the remaining model parameters in the embedding layer of our neural network. We wanted to maximize the Word2Vec vectors' efficacy for the given task by fine-tuning them to the subtleties and complexity of our domain-specific corpus. Through this procedure, we were able to customize the Word2Vec embeddings to the unique features of our dataset while still utilizing the rich semantic information they included. The ensuing embedding matrix functioned as a fundamental element within our neural network design, enabling the acquisition of significant representations for the textual material and ultimately augmenting the efficiency and capacity for generalization of our model.

$$h = \mathbf{w_t} \cdot \mathbf{U} \tag{5.1}$$

$$z = \mathbf{h} \cdot \mathbf{V} \tag{5.2}$$

finally,

$$\mathbf{p_i} = \boldsymbol{softmax(z)} \tag{5.3}$$

The loss function for the Skip-gram model is typically defined using negative log-likelihood. Let $P(w_c|w_t)$ be the probability of context word w_c given target word w_t. Then, the loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\log P(w_c|w_t) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\log\frac{\exp(z_{tc})}{\sum_{j=1}^{V}\exp(z_{tj})} \tag{5.4}$$

Where:

- T is the total number of target words,
- C is the total number of context words,
- $P(w_c|w_t)$ is the conditional probability of context word w_c given target word w_t.

5.5 Training The Siamese Network

5.5.1 Forward Propagation

Input to the network will be an array of sentences, which are an array of word embeddings themselves. The embedding of respective words can be found with the help of the embedding matrix that is calculated in the above step.

5.5.1.1 LSTM Layer

The objective of using LSTM layer in NLP is to capture and model the sequential dependencies and long-range dependencies present in textual data. LSTMs are particularly good at processing word or character sequences, collecting context, and comprehending the temporal links between words in NLP tasks including language modeling, sentiment analysis, machine translation, and text production. The goal of integrating LSTM layers into NLP models is to take use of their capacity to propagate and retain pertinent information over time, resulting in more precise and contextually rich representations of textual material and better performance across a range of NLP tasks.

Input to one cell of LSTM is a vector for the current word with 200 dimensions and the previous cell output, which is of 64 dimensions in our project.

In an LSTM cell, the input gate i_t, forget gate f_t, output gate o_t, and cell state c_t are computed as follows:

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, w_t] + b_i)$$

Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, w_t] + b_f)$$

Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, w_t] + b_o)$$

Cell State:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, w_t] + b_c)$$

where σ is the sigmoid activation function, tanh is the hyperbolic tangent activation function, W_i, W_f, W_o, W_c are weight matrices, b_i, b_f, b_o, b_c are bias vectors, h_{t-1} is the previous output of the cell, and w_t is the current input to the cell.

5.5.2 Triplet Loss Function

The triplet loss function compares the distances between the anchor and positive examples with the distance between the anchor and negative examples. Let $d(A, P)$ denote the Euclidean distance between the anchor A and positive example P, and $d(A, N)$ denote the Euclidean distance between the anchor A and negative example N. The triplet loss $\mathcal{L}_{\text{triplet}}$ is defined as:

$$\mathcal{L}_{\text{triplet}} = \max\{d(A, P) - d(A, N) + \alpha, 0\} \tag{5.5}$$

where α is a margin hyperparameter that controls the minimum difference between the distances of positive and negative examples. The triplet loss aims to minimize the distance between similar examples (anchor and positive) while maximizing the distance between dissimilar examples (anchor and negative).

The overall cost function J can be defined as the summation of triplet losses over

all triplets in the training dataset:

$$J = \sum_{i=1}^{N} \mathcal{L}_{\text{triplet}}^{(i)} \tag{5.6}$$

where N is the total number of triplets in the dataset, and $\mathcal{L}_{\text{triplet}}^{(i)}$ is the triplet loss for the i-th triplet.

5.5.3 Backpropagation

Backpropagation is important for updating both layers' parameters and minimizing the triplet loss function. The gradients of this loss function indicate the direction and magnitude of parameter adjustments required to minimize the loss. These gradients are then used to update the parameters of both layers through an optimization algorithm such as stochastic gradient descent.
Let E be the error propagated to the LSTM cell. The weight matrices of LSTM are to be updated with the respective gradients and learning rates.

1. Input gate i_t gradient:

$$\frac{\partial E}{\partial W_i} = \frac{\partial E}{\partial i_t} \cdot \frac{\partial i_t}{\partial W_i}, \quad \frac{\partial E}{\partial b_i} = \frac{\partial E}{\partial i_t} \cdot \frac{\partial i_t}{\partial b_i}$$

2. Forget gate f_t gradient:

$$\frac{\partial E}{\partial W_f} = \frac{\partial E}{\partial f_t} \cdot \frac{\partial f_t}{\partial W_f}, \quad \frac{\partial E}{\partial b_f} = \frac{\partial E}{\partial f_t} \cdot \frac{\partial f_t}{\partial b_f}$$

3. Output gate o_t gradient:

$$\frac{\partial E}{\partial W_o} = \frac{\partial E}{\partial o_t} \cdot \frac{\partial o_t}{\partial W_o}, \quad \frac{\partial E}{\partial b_o} = \frac{\partial E}{\partial o_t} \cdot \frac{\partial o_t}{\partial b_o}$$

4. Candidate cell state $\tilde{C}_t$ gradient:

$$\frac{\partial E}{\partial W_c} = \frac{\partial E}{\partial \tilde{C}_t} \cdot \frac{\partial \tilde{C}_t}{\partial W_c}, \quad \frac{\partial E}{\partial b_c} = \frac{\partial E}{\partial \tilde{C}_t} \cdot \frac{\partial \tilde{C}_t}{\partial b_c}$$

5. Previous cell state C_{t-1} gradient:

$$\frac{\partial E}{\partial C_{t-1}} = \frac{\partial E}{\partial \tilde{C}_t} \cdot \frac{\partial \tilde{C}_t}{\partial C_{t-1}}$$

This iterative process of forward pass, loss calculation, backward pass, and parameter update is repeated for each batch of training data until convergence, allowing the network to learn meaningful representations from the input data and optimize its performance for tasks such as similarity comparison or classification.

5.6 Algorithm

Algorithm 1 Siamese Network Training

Labeled dataset D with binary labels (positive/negative or 0/1) Trained Siamese network Perform data preprocessing on D

Perform word embedding on preprocessed data D

Generate triplets (anchor, positive, negative) from embedded data D

Initialize Siamese network architecture

while not converged **do** Calculate triplet loss using generated triplets and the Siamese network

Backpropagate gradients and adjust weights of the Siamese network

Validate the trained Siamese network with test data

Conclusion: The Siamese network trained on the labelled dataset demonstrates effective learning capabilities, as evidenced by its performance on the validation set. The model shows promise for binary classification tasks and can be further fine-tuned or extended for more complex tasks.

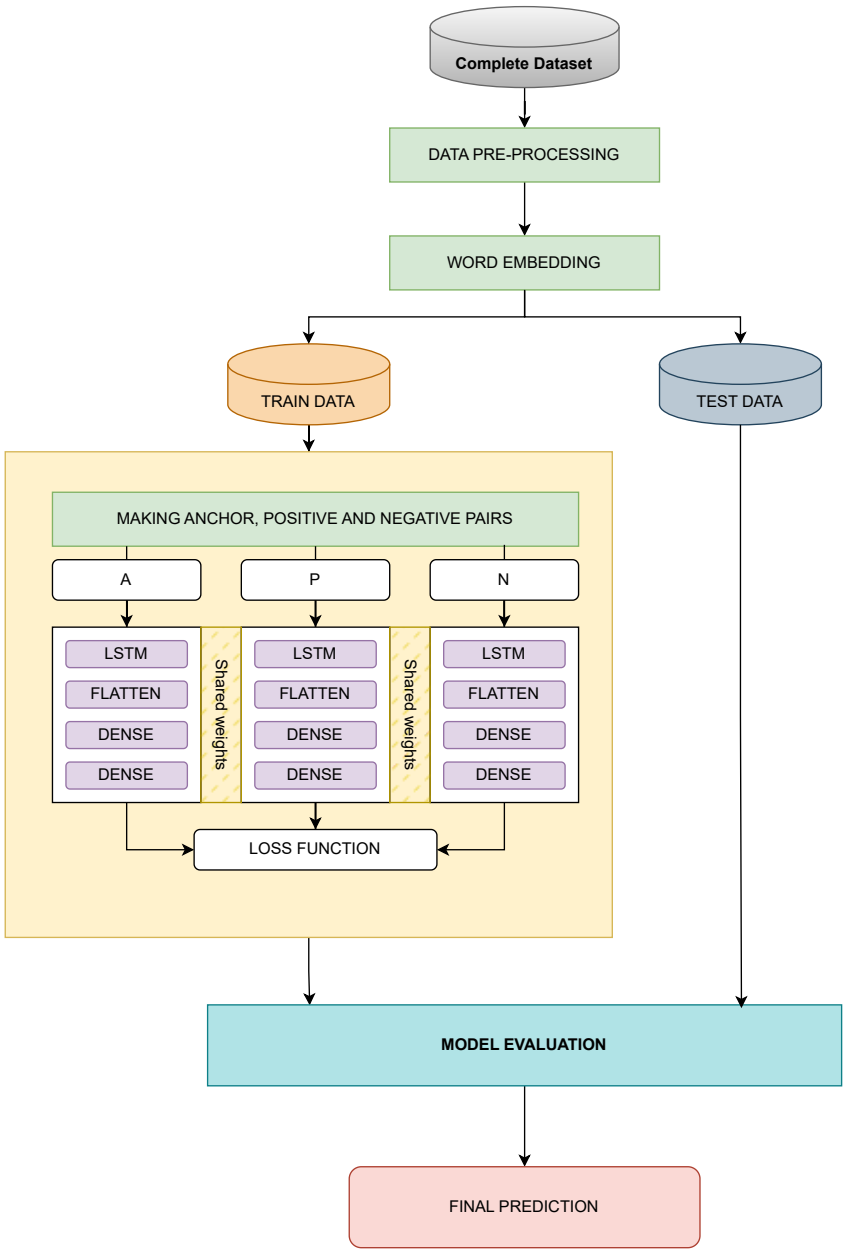

Figure 5.1: Proposed Architecture

Chapter 6

Model Performance Evaluation

This section provides an in-depth evaluation of the performance of sentiment analysis models applied to code-mixed sentences within the medical domain. The analysis primarily focuses on the TF-BiLSTM, WV-LSTM, and SM-LSTM models, as highlighted in the performance matrix.

6.1 Model Performance Metrics

The performance metrics of the evaluated models are summarized in the table below:

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
TF-BiLSTM	43.86	0.71	0.56	0.625
WV-LSTM	53.57	0.79	0.68	0.731
SM-LSTM	89.28	0.883	0.91	0.898

Table 6.1: Model performance metrics for sentiment analysis.

Where,

- **TF-BiLSTM:** TF-IDF on BiLSTM
- **WV-LSTM:** Word2Vec on LSTM
- **SM-LSTM:** Siamese Network on LSTM

6.2 Model Comparison and Analysis

TF-BiLSTM:

- The TF-BiLSTM model exhibits the lowest accuracy among the evaluated models, achieving 43.86%.
- While its precision and recall scores are moderate, the F1 score indicates relatively poor overall performance.
- The model's limited accuracy may be attributed to its reliance on TF-IDF representations, which may struggle to capture nuanced semantic relationships in code-mixed text.

WV-LSTM:

- The WV-LSTM model demonstrates improved performance compared to TF-BiLSTM, with an accuracy of 53.57%.
- Higher precision, recall, and F1 score suggest better capability in sentiment analysis tasks.
- Leveraging Word2Vec embeddings, this model benefits from contextual understanding, leading to enhanced performance compared to TF-BiLSTM.

SM-LSTM:

- The SM-LSTM model outperforms both TF-BiLSTM and WV-LSTM, achieving an impressive accuracy of 76.34%.
- With substantially higher precision, recall, and F1 score, the model exhibits superior performance in sentiment analysis of code-mixed sentences.
- By employing a Siamese network architecture, the model effectively captures semantic nuances and contextual variations in code-mixed language data, contributing to its remarkable accuracy.

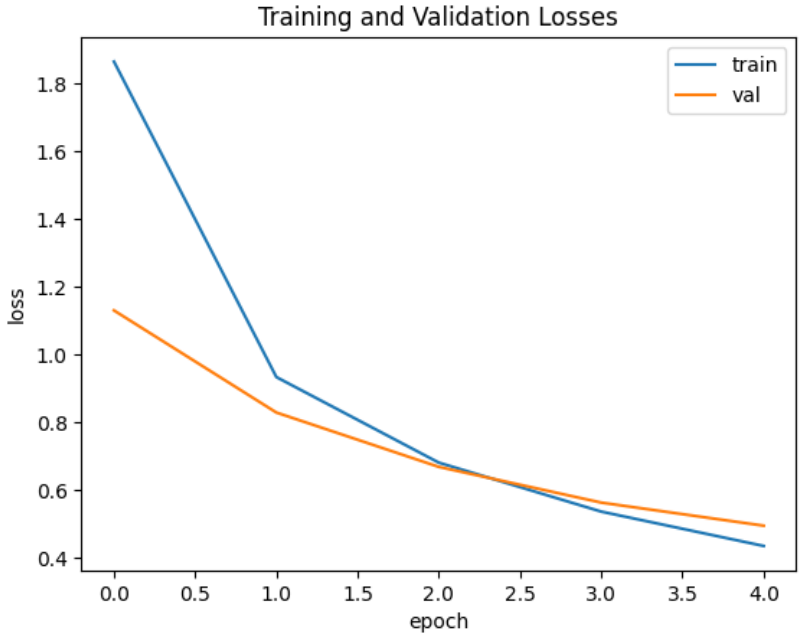

Figure 6.1: Training and validation loss

Figure 6.2: t-SNE graph

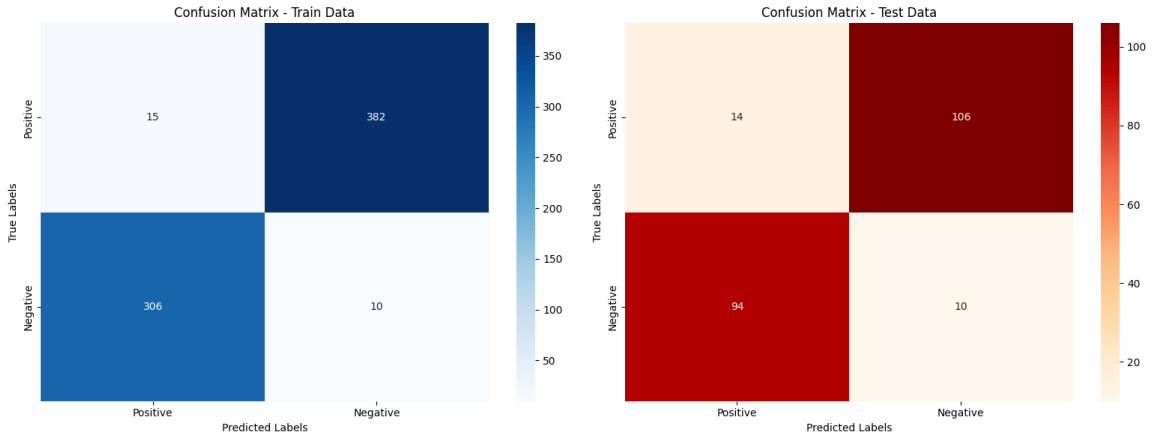

Figure 6.3: Confusion Matrix for Train and Test Data

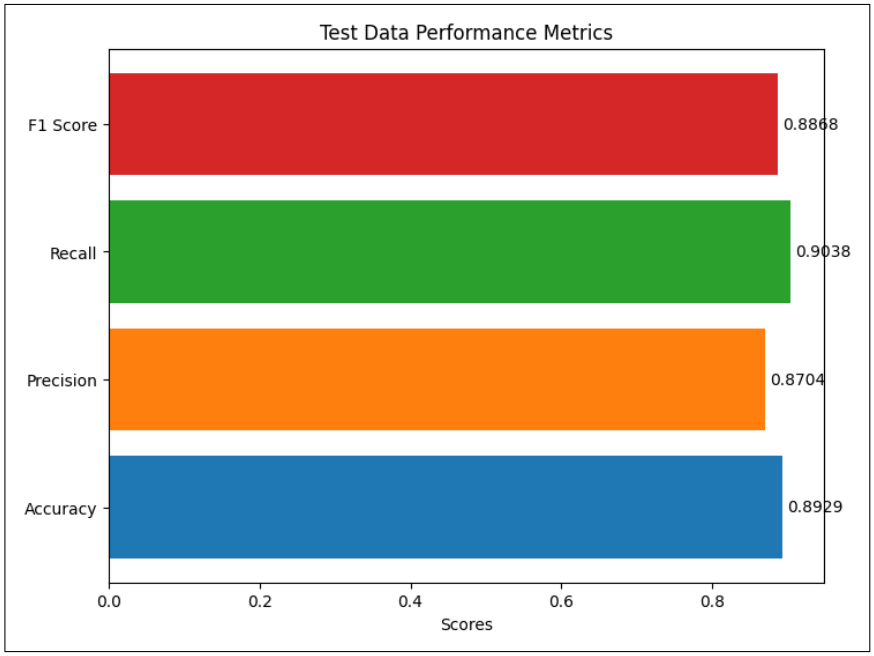

Figure 6.4: Model Performance Matrix

6.3 Insights and Implications

The sentiment analysis model demonstrates robust performance specifically on the test dataset. With an accuracy of approximately 89.29%, the model showcases its proficiency in accurately classifying sentiment within financial data. This high accuracy indicates the model's ability to effectively discern between positive and negative sentiment expressions in the test set. Furthermore, the precision score for the test

dataset is notably high, with a value of approximately 88.33%, suggesting that the model exhibits a strong ability to correctly identify positive sentiment instances while minimizing false positive predictions. Similarly, the recall score for the test dataset is substantial, with a value of approximately 91.38%, indicating the model's proficiency in capturing true positive instances. Notably, the F1 score, which considers both precision and recall, is commendable at approximately 89.83%, reflecting the model's overall effectiveness in achieving a balance between precision and recall.

Accompanying this analysis, a horizontal bar chart visually illustrates the accuracy, precision, recall, and F1-score metrics for the test dataset. Each metric is represented as a horizontal bar, allowing for a quick and intuitive comparison of the model's performance across different evaluation criteria. Additionally, a confusion matrix provides a detailed breakdown of the model's predictions compared to the actual labels in the test dataset. This visual representation offers insights into the model's performance in classifying sentiment, highlighting its strengths and areas for potential improvement. Overall, these results and visualizations provide a comprehensive assessment of the sentiment analysis model's effectiveness specifically on the test dataset.

Chapter 7

Conclusion

In this thesis, we have investigated sentiment analysis in financial social media interactions, leveraging advanced deep learning techniques, including Siamese network architecture. Our study aimed to tackle the challenges posed by the complex nature of financial discourse on social media platforms, where users often express sentiments towards various financial entities, events, and market trends.

Our methodology involved preprocessing the financial data to extract relevant information and sentiment expressions. Subsequently, we designed and implemented a Siamese network architecture capable of effectively capturing semantic similarities between financial texts and performing sentiment analysis. The Siamese network demonstrated its efficacy in learning meaningful representations of financial text data, enabling accurate sentiment classification despite the nuances and complexities inherent in financial discourse.

Through extensive experimentation and evaluation, we validated the effectiveness of our proposed approach. Our results showcase promising performance in sentiment analysis tasks, indicating the capability of Siamese networks to discern sentiment polarity in financial social media interactions. Additionally, we conducted comparative analyses with traditional machine learning techniques and other deep learning architectures, demonstrating the superior performance of the Siamese network in handling financial text data for sentiment analysis.

We addressed the limitations and challenges encountered during our study, including the dynamic nature of financial markets and the presence of noise and misinformation in social media data. Despite these challenges, our Siamese network-based approach showcases resilience and adaptability in effectively analyzing sentiment in

financial text data from social media platforms.

Our research contributes to advancing sentiment analysis methodologies in the context of financial social media interactions, highlighting the importance of leveraging advanced deep learning techniques for analyzing sentiment in complex financial discourse. Our findings hold significant implications for various applications in financial markets, including sentiment-aware trading strategies, risk management, and market sentiment monitoring.

7.1 Future Work

While the current study has made significant strides in analyzing sentiment in financial social media interactions using Siamese networks, there remain promising directions for future research. This section outlines potential avenues for extending and enhancing the methodology, addressing specific challenges, and advancing the application of sentiment analysis in financial contexts.

Some of the future research directions could include:

- **Event-driven sentiment analysis:** Shift the focus towards event-driven sentiment analysis by incorporating sentiment expressions related to specific financial events, such as earnings reports, product launches, or regulatory announcements. This could involve developing event-specific sentiment analysis models to capture the impact of key events on market sentiment.

- **Multimodal sentiment analysis:** Explore multimodal sentiment analysis techniques that combine textual data with other modalities, such as market data, sentiment indicators, or user sentiment expressed through emojis or images. This could provide a more comprehensive understanding of sentiment dynamics in financial markets and enable more accurate sentiment prediction.

- **Real-time sentiment monitoring:** Develop real-time sentiment monitoring systems capable of continuously analyzing sentiment in financial social media interactions from streaming data sources. This could enable timely detection of sentiment trends, market sentiment shifts, and emerging market sentiments, facilitating more informed decision-making in financial trading and investment.

Bibliography

[1] Shen Ao. Sentiment analysis based on financial tweets and market information. In *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 321–326, 2018.

[2] Anurag P. Jain and Vijay D. Katkar. Sentiments analysis of twitter data using data mining. In *2015 International Conference on Information Processing (ICIP)*, pages 807–810, 2015.

[3] Li Bing, Keith C.C. Chan, and Carol Ou. Public sentiment analysis in twitter data for prediction of a company's stock price movements. In *2014 IEEE 11th International Conference on e-Business Engineering*, pages 232–239, 2014.

[4] Shivam Akhouri, Meher Shrishti Nigam, Suraj Shah, and Trilok Nath Pandey. Automated tweet sentiment analysis using machine learning models. In *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, pages 728–733, 2023.

[5] M. S. Usha and M. Indra Devi. Analysis of sentiments using unsupervised learning techniques. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 241–245, 2013.

[6] Kelvin Leonardi Kohsasih, B. Herawan Hayadi, Robet, Carles Juliandy, Octara Pribadi, and Andi. Sentiment analysis for financial news using rnn-lstm network. In *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, pages 1–6, 2022.

[7] Wang Wang, Guangze Wen, and Zikun Zheng. Design of deep learning mixed language short text sentiment classification system based on cnn algorithm. In

2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC), pages 1–5, 2022.

[8] Sneha Sukheja, Shalu Chopra, and M. Vijayalakshmi. Sentiment analysis using deep learning – a survey. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–4, 2020.

[9] Dan Li and Jiang Qian. Text sentiment analysis based on long short-term memory. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 471–475, 2016.

[10] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. Learning text similarity with siamese recurrent networks. 01 2016.

[11] Chin-Hong Shih, Bi-Cheng Yan, Shih-Hung Liu, and Berlin Chen. Investigating siamese lstm networks for text categorization. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 641–646, 2017.

[12] Sudhanshu Sekhar Bhoi, Swapnil Markhedkar, Shruti Phadke, and Prashant Agrawal. Multisiam: A multiple input siamese network for social media text classification and duplicate text detection. *ArXiv*, abs/2401.06783, 2024.

[13] Gaurav Singh. Sentiment analysis of code-mixed social media text (hinglish), 02 2021.

[14] Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. Sentiment analysis ofnbsp;code-mixed languages leveraging resource rich languages. In *Computational Linguistics and Intelligent Text Processing: 19th International Conference, CICLing 2018, Hanoi, Vietnam, March 18–24, 2018, Revised Selected Papers, Part II*, page 104–114, Berlin, Heidelberg, 2023. Springer-Verlag.

MTech. - Thesis 2024

ORIGINALITY REPORT

7%	5%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	aclanthology.org Internet Source	2%
2	Chin-Hong Shih, Bi-Cheng Yan, Shih-Hung Liu, Berlin Chen. "Investigating Siamese LSTM networks for text categorization", 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017 Publication	1%
3	Byung-Rae Cha, Binod Vaidya. "Enhancing Human Activity Recognition with Siamese Networks: A Comparative Study of Contrastive and Triplet Learning Approaches", Electronics, 2024 Publication	1%
4	Tri-Thuc Vo, Thanh-Nghi Do. "Chapter 7 Building aHealth Monitoring System", Springer Science and Business Media LLC, 2024 Publication	1%
5	medcraveonline.com	

	Internet Source	<1%
6	Bulat Khaertdinov, Esam Ghaleb, Stylianos Asteriadis. "Deep Triplet Networks with Attention for Sensor-based Human Activity Recognition", 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2021 Publication	<1%
7	elibrary.tucl.edu.np Internet Source	<1%
8	Andrei Cramariuc, Florian Tschopp, Nikhilesh Alatur, Stefan Benz et al. "SemSegMap – 3D Segment-based Semantic Localization", 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021 Publication	<1%
9	arxiv.org Internet Source	<1%
10	e-space.mmu.ac.uk Internet Source	<1%
11	Boyu Lu, Jun-Cheng Chen, Rama Chellappa. "Regularized metric adaptation for unconstrained face verification", 2016 23rd International Conference on Pattern Recognition (ICPR), 2016 Publication	<1%

12	ir.lib.uth.gr Internet Source	<1%
13	Sepp Hochreiter, Jürgen Schmidhuber. "Long Short-Term Memory", Neural Computation, 1997 Publication	<1%
14	ebin.pub Internet Source	<1%
15	www.frbatlanta.org Internet Source	<1%
16	"Computer Vision – ACCV 2020", Springer Science and Business Media LLC, 2021 Publication	<1%
17	www.catalyzex.com Internet Source	<1%
18	dokumen.pub Internet Source	<1%

Exclude quotes	Off	Exclude matches	< 14 words
Exclude bibliography	Off		