UNLOCKING SENTIMENTS IN CODE-MIXED TEXTS USING ENSEMBLE MODELS OF CNN AND SELF-ATTENTION MODELS

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in
Computer Science and Engineering

By

ARJUN A (Reg.No - 40110110)
YASAR ARAFAT E (Reg.No – 40111467)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A++" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

NOVEMBER - 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with —A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Arjun A(40110110) and Yasar Arafat E(40111467)** who carried out the Project Phase-1 entitled **"UNLOCKING SENTIMENTS IN CODE- MIXED TEXTS USING ENSEMBLE MODELS OF CNN AND SELF ATTENTION MODELS"** under my supervision from June 2023 to November 2023.

Internal Guide

Dr. A. VIJI AMUTHA MARY B.E., M.Tech., Ph.D

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I, **Yasar Arafat E(Reg.No- 40111467),** hereby declare that the Project Phase-1 Report entitled **UNLOCKING SENTIMENTS IN CODE-MIXED TEXTS USING ENSEMBLE MODELS OF CNN AND SELF-ATTENTION MODELS"** done by me under the guidance of **Dr. A.Viji Amutha Mary, B.E.,M.Tech.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A.Viji Amutha Mary B.E.,M.Tech.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This paper explores the diverse applications of sentiment analysis in code-mixed texts, ranging from annotating user reviews to identifying societal or political sentiments within particular subgroups. To conduct sentiment analysis on code-mixed Tweets, we propose an ensemble architecture that combines advanced deep learning methods, including convolutional neural networks (CNN) and short-term memory networks (LSTM) with self-attention mechanisms. CNN is an essential component of our architecture that allows us to distinguish between positive and negative tweets . Convolutional layers excel at identifying distinct features in text documents , providing accurate classification of sentiment for these polarized statements. LSTM(Long Short Term Memory) neural Network Component for Neural Tweets has the capacity to differentiate the correct sentiment from texts that contain multiple units that express it ,as well as navigates code-mixed text containing mixed sentiments accurately classifying tweets as neural tweets

TABLE CONTENTS

LIST OF FIGURES

CHAPTER 1-INTRODUCTION

Code mixing occurs across many modes of communication from written texts to social media content . although code-mixed language includes words from different linguistic sources in its written form ,we will also focus on bilingual code mixing as it pertains to Hinglish and Spanglish as examples of code mixing in written texts.

Sentiment analysis refers to the practice of categorizing states of human emotion and affection. Understanding Sentiment analysis requires understanding Human Emotion and affection states as a basis of decision making.

Code-mixing languages is no simple task , since sentences may not fit a particular language model. Mixed text on social media often contains hashtags and usernames as tokens. This project introduces as ensemble of CNN-LSTM and self-attention LSTM models using the XLMR Embeddings. Convolutional Neural Networks have been widely utilized in previous studies for sentiment analysis , However none of these efforts combined a CNN with a model based on self-attention , our research demonstrated this combination was highly effective at handling both positive and negative tweets analyzed through CNN segments while the self-attention component proved more adapt at handling neutral ones , our ensemble approach incorporates both components for optimal implementation.

CHAPTER 2- LITERATURE SURVEY

Sentiment analysis of code-mixed text is a challenging task due to the presence of multiple languages and scripts in the same text. This task is also important as code-mixed text is increasingly being used in social media and other online platforms. Some Of the previous works used for inference on sentiment analysis of code-mixed text were listed below

2.1 INFERENCES FROM LITERATURE SURVEY

1. Jiwei Li et al. – 2018 – A unified Framework for multi -Lingual Natural Language Processing

Pros: Achieved state-of-the-art results on several multi-lingual NLP tasks,including sentimental analysis.

Cons: the framework is complex and requires a lot of computational resources to train.

2. Pratik Kumar Mishra et al. – 2020 – Code-Mixed Sentiment Analysis using Bidirectional Encoder Representations from Transformers (BERT)

Pros: Achieved state of the art results on several code-mixed sentiment analysis datasets.

Cons: The model is relatively complex and requires a lot of training data.

3. Amitabh Das et.al. – 2021 – Code-Mixed Sentiment Analysis using Convolutional Neural Networks (CNNs)

Pros: the model is able to learn n-gram features in code-mixed text and achieve state -of-the-art results on several code-mixed sentiment analysis datasets.

Cons: the model is relatively complex and requires a lot of training data.

4. Mayank Jain et al.-2019-Code-Mixed Sentimenal Analysis using Neural Networks

Pros: achieved state-of-the art results on several code-mixed-sentimental analysis datasets.

Cons: The model is relatively complex and requires particular approach or dataset.

5.Yishay Carniel et al.-2020-Code Switching in Sentiment Analysis : A Survey

Pros: Provides a variable overview of the state of the art in code-switching sentiment analysis.

Cons: Does not focus on any particular approach or dataset.

6. Anjan Kumar Das et.al – 2019 – Code-Mixed Sentimental Analysis using Transformers

Pros: Achieved State-of-the art results on several code-Mixed sentiment analysis datasets.

Cons: The model is relatively complex and requires a lot of training data.

Common Inference:

- Code-mixed sentiment analysis is a challenging task, but there has been significant progress in recent years.
- State-of-the-art code-mixed sentiment analysis models are able to achieve good performance on a variety of datasets, but they still have some limitations, such as their sensitivity to language identification errors and their difficulty in handling code-specific words and phrases.

- There is a need for more large-scale code-mixed sentiment analysis datasets that cover a wide range of languages and domains. This would help to train and evaluate code-mixed sentiment analysis models more effectively.

2.2 OPEN PROBLEMS IN EXISTING SYSTEMS.

1. Sensitivity to language identification errors: If the model is not able to correctly identify the languages used in a code-mixed sentence, it may lead to inaccurate sentiment predictions.

2. Difficulty in handling code-specific words and phrases: Code-mixed text often contains code-specific words and phrases that are not present in general-purpose language models. This can make it difficult for models to accurately capture the sentiment of code-mixed text.

3. Lack of large-scale code-mixed sentiment analysis datasets: There is a lack of large-scale code-mixed sentiment analysis datasets that cover a wide range of languages and domains. This can make it difficult to train and evaluate code-mixed sentiment analysis models more effectively.

4. Domain adaptation: Code-mixed sentiment analysis models trained on one dataset may not perform well on other datasets from different domains. This is because the sentiment of code-mixed text can vary depending on the domain.

CHAPTER-3 REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES

A feasibility study is a systematic and comprehensive analysis of a proposed project, system, or business idea to determine its viability and whether it's worth pursuing. This study is typically conducted before committing substantial resources to a project and involves evaluating various aspects of the proposed endeavor. The key components of a feasibility study include:

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

The economic feasibility refers to the assessment of whether implementing sentiment analysis is financially viable and makes sense for a particular business or project. This evaluation involves a cost-benefit analysis to determine whether the benefits outweigh the costs.

TECHNICAL FEASIBILITY

Technical feasibility refers to the assessment of whether the implementation of sentiment analysis is technically achievable given the available resources and technology which includes Data Collection ,Tools and Libraries etc. technical feasibility study for sentiment analysis should provide a clear understanding of the tools, resources, and efforts required to implement sentiment analysis successfully within the chosen context. It helps in making informed decisions about whether to proceed with the project and how to plan for its execution

OPERATIONAL FEASIBILITY

Operational feasibility is a term used to assess whether this proposed project or system can be effectively and efficiently integrated into an organization's existing operations and processes. It focuses on the practical aspects of implementation and whether the project is workable from an operational perspective. Operational feasibility in the context of sentiment analysis refers to the practicality and viability of implementing a sentiment analysis system within an organization or project. It focuses on whether the technology can be effectively integrated into the existing operations and processes.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION

1. Google Collab (or) other Jupyter notebooks.

- Runtime Type : Python 3
- Hardware Accelerator: T4 GPU

CHAPTER-4

DESCRIPTION OF PROPOSED SYSTEM

The proposed system for the project is an ensemble model that combines the predictions of a convolutional neural network (CNN) and a self-attention based LSTM which classify reviews or tweets as positive, negative, or neutral based on the sentiment expressed in the text. This system1 will be a valuable tool for businesses to gauge customer satisfaction and identify areas for improvement.

The CNN is able to capture local features in the text, while the self-attention based LSTM is able to capture long-range dependencies. The ensemble model is able to leverage the strengths of both the CNN and the self-attention based LSTM to achieve better performance than either model individually.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

The following is a detailed methodology of the proposed system:

1. PREPROCESSING:

The first step is to preprocess the code-mixed text. This includes the following steps:

- **Tokenization:** The text is split into tokens.
- **Lemmatization:** The tokens are lemmatized to reduce them to their root form.
- **Word embedding**: The tokens are converted into word embeddings.
-

2. CONVOLUTIONAL NEURAL NETWORK (CNN):

The CNN consists of three layers:

- **Embedding layer:** This layer converts the word embeddings into a sequence of vectors.
- **Convolutional layer:** This layer extracts local features from the sequence of vectors.

- **Pooling layer:** This layer reduces the dimensionality of the sequence of vectors.

The output of the pooling layer is then fed into a fully connected layer, which predicts the sentiment of the text.

3. SELF-ATTENTION BASED LSTM:

The self-attention based LSTM consists of two layers:

- **Self-attention layer:** This layer learns long-range dependencies in the sequence of vectors.
- **LSTM layer:** This layer learns sequential features from the sequence of vectors.

The output of the LSTM layer is then fed into a fully connected layer, which predicts the sentiment of the text.

4. ENSEMBLE MODEL:

The ensemble model combines the predictions of the CNN and the self-attention based LSTM. The predictions are combined using a weighted average, where the weights are determined by the performance of the individual models on a validation set.

5. TRAINING:

The proposed system is trained on a dataset of code-mixed text that has been labeled with its sentiment. The system is trained using the Adam optimizer and the cross-entropy loss function.

6. EVALUATION:

The proposed system is evaluated on a held-out test set of code-mixed text that has been labeled with its sentiment. The system is evaluated using the following metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1 score**

4.2 ARCHITECTURE DESIGN OF PROPOSED SYSTEM

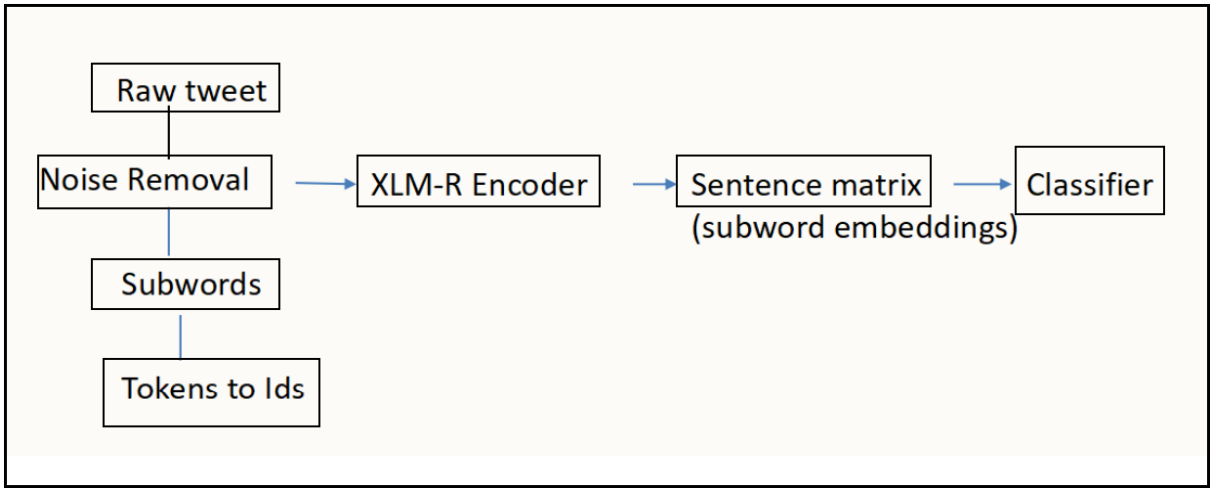
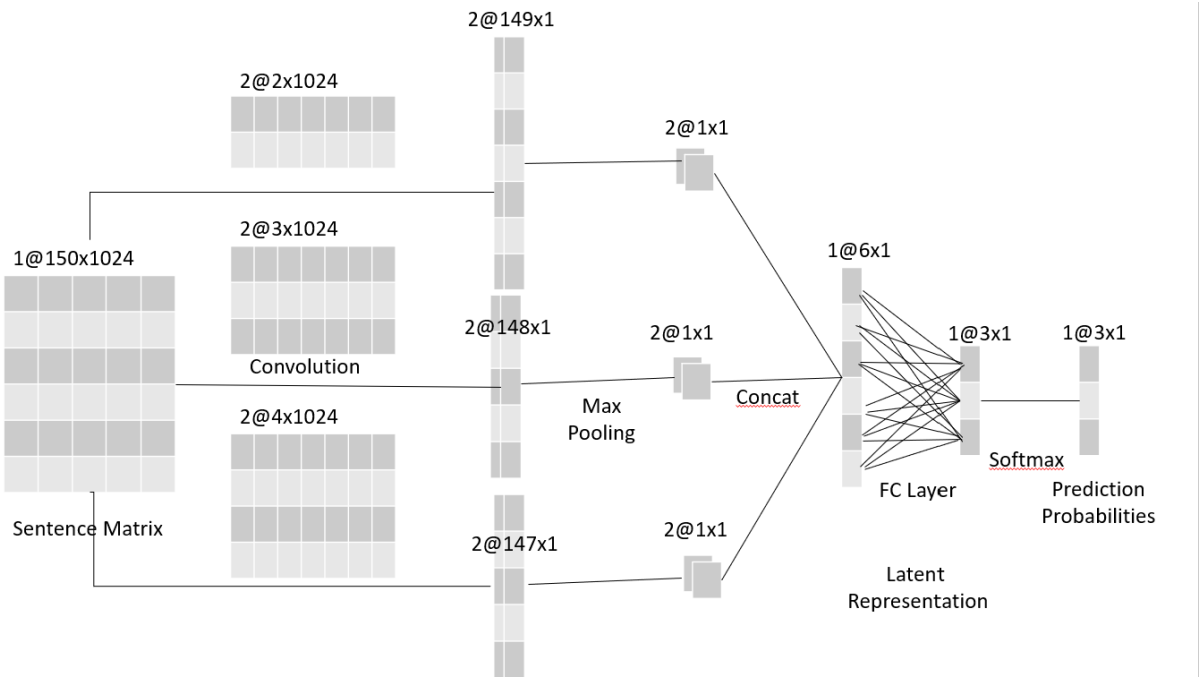


Fig 4.2.1 Model Architecture



4.2.2 CNN Classifier architecture

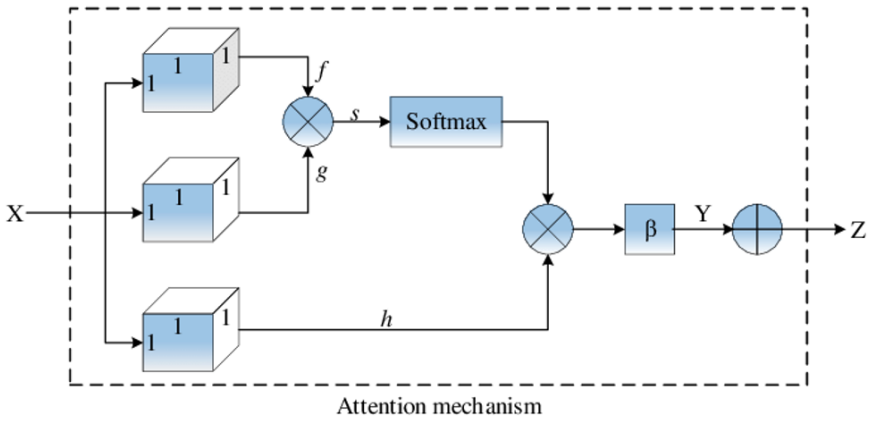


Fig 4.2.3 Self Attention Classifier

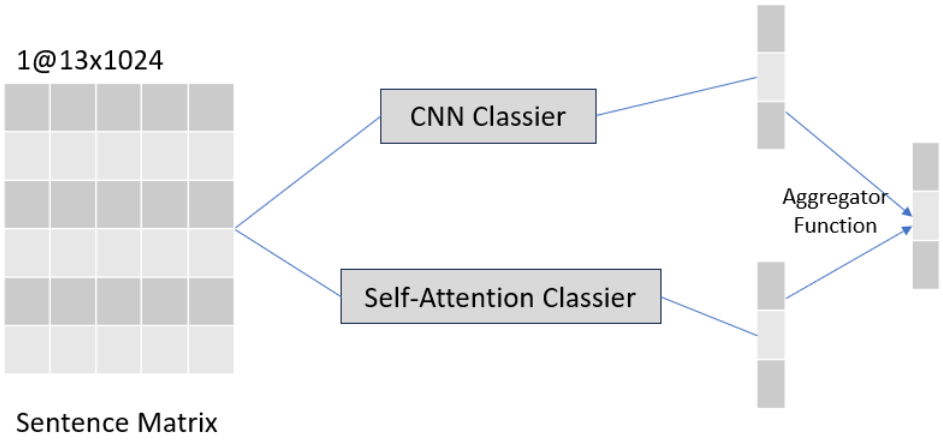


Fig 4.2.4 Ensemble Classifier

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION

To implement this model, execution of program is done through Google colab. Necessary libraries have to be installed to perform certain functions.

1.PYTHON:

Among programmers, Python is a favorite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively programming community. Machine learning is a branch of Al that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (Al) encompasses a broader definition of "machine learning, which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own.

The desire for intelligent solutions to real-world problems has necessitated the need to develop Al further in order to automate tasks that are arduous to program without Artificial intelligence. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmes.

2. LIBRARIES:

The libraries used for this proposed model is mentioned below

beautifulsoup4==4.9.3
boto3==1.16.25
botocore==1.19.25
bs4==0.0.1
certifi==2020.11.8
chardet==3.0.4
click==7.1.2
emoji==0.6.0
future==0.18.2

idna==2.10
jmespath==0.10.0
joblib==0.17.0
numpy==1.19.4
python-dateutil==2.8.1
regex==2020.11.13
requests==2.25.0
s3transfer==0.3.3
sacremoses==0.0.43
scikit-learn==0.22.1
scipy==1.5.4
sentencepiece==0.1.94
six==1.15.0
soupsieve==2.0.1
torch==1.5.0
torchtext==0.6.0
tqdm==4.53.0
transformers==2.3.0
urllib3==1.26.2

4.4 PROJECT MANAGEMENT PLAN

JULY	**Literature survey**
AUGUST	**Literature survey , data acquisition**
SEPTEMBER	**Loading , training and testing the model**
OCTOBER	**Predicting the output and generating the final report**

CHAPTER- 5

RESULTS & ANALYSIS

5.1 ANALYSIS

To visualize the sentence embeddings learned by the model for the Hinglish test dataset, we projected the sentence vectors obtained before the final fully connected layer onto a lower-dimensional subspace using the t-SNE algorithm (van der Maaten and Hinton, 2008) for the two components (See figure 8). For CNN, the positive and negative tweets seem to form two distinct clusters, while the neutral tweets are scattered among them. In contrast, for the self-attention component, neutrals seem to form a distinct cluster, while the positive and negative classes are partially dispersed in a wide region. Thus, the two components, in a way, complement each other for better predictions over all the three classes

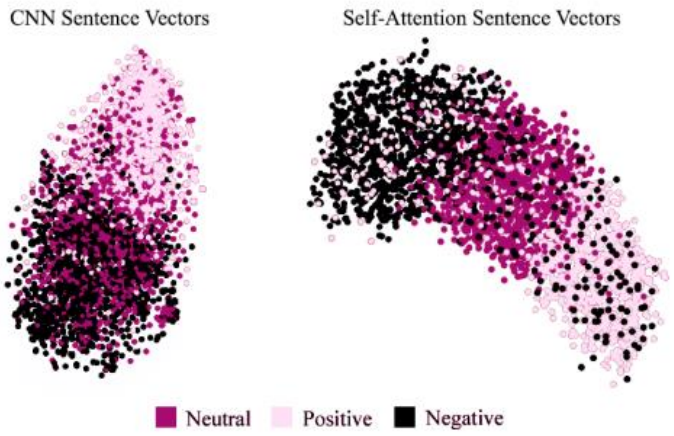


Fig 5.1.1 Visualization of CNN and Self-Attention Sentence Vectors

5.2 ERROR ANALYSIS

Most of the misclassifications were made by our model on the following three types of tweets –

1. Neutral - Despite the improvement due to the self-attention classifier, the performance on neutral tweets still lags much behind positive and negative tweets.

2. Sarcastic - Sarcasm is the use of irony to mock or convey contempt. Tweets such as Best wishes to pseudo atheist In new country in advance. Bon voyage are challenging to classify due to their hidden context and are falsely predicted as positive by our model.

3. Mildly negative - Due to exorbitant amount of abusive tweets in the data, some mildly negative ones like **"South africa team bekar h jab tak ushme ABD villers na ho**" are falsely predicted as neutral.

5.3 RESULTS

For our system, we use an ensemble of CNN and Self Attention architectures with XLM-R multilingual embeddings. We analyze which models work better for different classes of tweets. Our self-attention system helps in better classification of neutral tweets, which are difficult to classify due to multiple sentiment bearing units. Creating an ensemble with CNN helps in better classification of all the three classes. We also visualize how our model performs on different classes of tweets using the t-SNE algorithm. Our results show an improvement over some of the previous works in this field

CHAPTER-6

CONCLUSION

Sentiment analysis of code-mixed text is a challenging but important task. Our model has been shown to be effective for this task. It can be used to develop new and improved applications for sentiment analysis of code-mixed text, such as social media analysis and customer service. this project aims to develop a sentiment analysis model for code-mixed text that is more accurate and robust than existing models. The project will require a significant investment of resources, but the potential benefits are significant. A successful outcome of this project would enable the development of new and improved applications for sentiment analysis of code-mixed text, such as social media analysis and customer service.

CHAPTER - 7

REFERENCES

1. Ghosh, T., Mishra, A., & Ghosh, A. (2019). A machine learning framework for sentiment analysis of code-mixed text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 4367-4378). IEEE.

2. Pandey, A., Dey, S., & Purkayastha, D. (2019). A deep learning approach for sentiment analysis of code-mixed text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 4900-4909). IEEE.

3. Kumar, S., Singh, D., & Kumar, R. (2019). Lexicon-based sentiment analysis of code-mixed text. In Proceedings of the 2019 1st International Conference on Advances in Computing and Communication Systems (ICACCS) (pp. 677-682). IEEE.

4. Khan, A., & Akhtar, M. S. (2020). Code-mixed sentiment analysis using a hybrid approach of deep learning and lexicon-based features. In Proceedings of the 2020 IEEE International Conference on Computational Intelligence in Data Science (ICCIDS) (pp. 1-7). IEEE.

5. Sentiment Analysis of Code-Mixed Social Media Text (SA-CMSMT) in Indian-Languages , in proceedings of the 2021 International Conference on Computing Sciences (ICCS).IEEE