

Enhancing Code-Mixed Sentiment Analysis: A Transformer-based Approach

A PROJECT REPORT

Submitted by

**Ridhima Handa (21BCS11228)
Vanshika Sharma (21BCS3782)
Vinayak Katoch (21BCS3806)**

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE WITH SPECIALIZATION IN
BIG DATA AND ANALYTICS**

Under the Supervision of:

Ms. Navjeet Kaur (e16069)



Chandigarh University, GHARUAN, MOHALI – 140413

PUNJAB

June 2024

BONAFIDE CERTIFICATE

Certified that this project report “Enhancing Code-Mixed Sentiment Analysis: A Transformer-based Approach” is the bonafide work of Ridhima Handa, Vanshika Sharma and Vinayak Katoch who carried out the project work under our supervision.

Mr. Aman Kaushik
HEAD OF THE DEPARTMENT
(AIT-CSE)

Ms. Navjeet Kaur
SUPERVISOR
(AIT-CSE)

Submitted for the project viva-voce examination held on 30 April,2024

INTERNAL EXAMINER

EXTERNAL EXAMINER

Abstract

The increasing presence of non-native English speakers on social media platforms has fuelled interest in analysing mood and emotion in regional languages and code-mixed data. Despite extensive research in sentiment and emotion analysis in English text, a notable gap exists in understanding English code-mixed texts. To address this, we propose an end-to-end transformer-based multitask framework designed for sentiment and emotion identification. Our approach includes the development of an emotion-annotated Hindi–English code-mixed dataset, facilitating research in this domain. Leveraging the pre-trained cross-lingual embedding model XLMR, our framework outperforms single-task and multitask baselines by integrating emotion recognition as an auxiliary task. This highlights its effectiveness in capturing nuanced emotional nuances within code-mixed content. Moreover, our model exhibits promise for various natural language processing (NLP) applications, showcasing robust performance even without ensemble techniques. By providing both a novel dataset and an effective methodology, our work aims to advance research in sentiment and emotion analysis, particularly in multilingual social media discourse.

Keywords: NLP (Natural Language Processing), XLMR (Cross-lingual Language Model Representation).

Table of Contents

Title Page	1
Abstract	3
1. Introduction	5-11
1.1 Problem Definition	5-7
1.2 Project Overview	7-10
1.3 Hardware Specification	10
1.4 Software Specification	11
2. Literature Survey	12-24
2.1 Existing System	12-17
2.2 Proposed System	17-20
2.3 Literature Review Summary	20-24
3. Problem Formulation	25-28
4. Objective	28-31
5. Methodologies	31-36
6. Experimental Setup	36-40
7. Result	41-50
8. Conclusion and Future Scope	50-55
9. Reference	55-60

1. Introduction

1.1 Problem Definition

With an emphasis on the understudied field of English code-mixed texts, this work fills a gap in the literature on sentiment analysis for code-mixed languages. The majority of the research that is currently available focuses on English, ignoring the particular difficulties presented by multilingual code-mixed data. We suggest a novel transformer-based method for sentiment analysis in order to close this gap. We address the lack of annotated data by generating an English code-mixed dataset with emotion annotations. Our end-to-end approach leverages transfer learning and is optimized on task-specific data using the XLMR cross-lingual embedding model. This method integrates emotion recognition as a multitask framework and improves sentiment detection as well. Our results highlight the effectiveness of this approach in handling the intricacies of code-mixed sentiment analysis by demonstrating its superiority over baseline models.

Our study addresses the research gap in sentiment analysis for English code-mixed texts, an area that has been largely overlooked in existing literature. While most studies focus on sentiment analysis in monolingual English, this work recognizes the unique challenges posed by multilingual code-mixed data and proposes a novel transformer-based method to address them. To overcome the scarcity of annotated data, the researchers generate a specialized English code-mixed dataset with emotion annotations. This dataset serves as a crucial resource for training and evaluating sentiment analysis models tailored specifically to code-mixed languages.

The proposed approach adopts an end-to-end methodology, leveraging transfer learning techniques and optimizing model performance using the XLMR cross-lingual embedding model. By fine-tuning the model on task-specific data, it enhances the model's ability to understand and interpret the complexities of sentiment expressions in code-mixed texts. This methodology is particularly relevant in the context of English code-mixed texts, where the interplay between English and other languages introduces additional challenges for sentiment analysis algorithms.

Our research aims to fill the gap in sentiment analysis for English code-mixed texts, an area that has received limited attention in existing literature. While most studies focus on analysing sentiments in monolingual English, our work acknowledges the unique challenges posed by multilingual code-mixed data. We propose a novel transformer-based approach to address these challenges. To mitigate the scarcity of annotated data, we create a specialized English code-mixed dataset with emotion annotations. This dataset serves as a valuable resource for training and evaluating sentiment analysis models specifically tailored to code-mixed languages

One of the key contributions of our work is the development of a multitask framework that integrates emotion recognition alongside sentiment detection. This holistic approach enables the model to capture a broader range of linguistic features and contextual cues, thereby improving its overall performance in code-mixed sentiment analysis tasks. Additionally, by leveraging transfer learning, our approach benefits from pre-existing knowledge captured by the XLMR model, which enhances its adaptability to code-mixed language environments.

Furthermore, our study highlights the importance of rigorous evaluation in assessing the efficacy of sentiment analysis models for code-mixed languages. By comparing our transformer-based approach against baseline models using established metrics such as F1-Score, accuracy, and precision-recall curves, we provide empirical evidence of its superior performance. This not only validates the effectiveness of our proposed method but also underscores the need for specialized approaches tailored to the unique characteristics of code-mixed texts.

Our research introduces a multitask framework that combines emotion recognition with sentiment detection. This comprehensive approach allows the model to encompass a wider array of linguistic features and contextual cues, leading to enhanced performance in code-mixed sentiment analysis tasks. Furthermore, leveraging transfer learning from the XLMR model enriches our approach's adaptability to code-mixed language contexts.

In conclusion, our work contributes to advancing the field of sentiment analysis by addressing the challenges associated with analyzing sentiment in English code-mixed texts. Through the development of a transformer-based approach, dataset curation, and rigorous evaluation, we demonstrate the feasibility and effectiveness of enhancing sentiment analysis in multilingual contexts. By providing insights and methodologies tailored for code-mixed languages, our study lays the foundation for future research in this emerging area of natural language processing.

1.2 Problem Overview

The work focuses on the developing field of English code-mixed texts and addresses the crucial problem of the scant progress in sentiment analysis for code-mixed languages. Even while sentiment analysis in English has been studied in

great detail, little is known about the difficulties presented by multilingual code-mixed data. The study presents a unique transformer-based method to improve sentiment analysis in order to close this gap. To overcome the lack of annotated data, an emotion-annotated English code-mixed dataset is curated. The suggested method blends emotion recognition into a multitask framework and enhances sentiment detection by utilizing transfer learning via the XLMR cross-lingual embedding model. The study highlights the transformational potential of the suggested paradigm in developing code-mixed sentiment and highlights the shortcomings in present techniques.

To tackle the scarcity of annotated data, the researchers compile an emotion-annotated Hindi–English (Hinglish) code-mixed dataset. This curated dataset serves as a valuable resource for training and evaluating sentiment analysis models tailored to code-mixed languages. The proposed method incorporates emotion recognition into a multitask framework, augmenting sentiment detection capabilities. Leveraging transfer learning through the XLMR cross-lingual embedding model further optimizes the model's performance on task-specific data. The study underscores the transformative potential of the proposed paradigm in advancing code-mixed sentiment analysis while also shedding light on the limitations of existing techniques.

The scarcity of annotated data poses a significant challenge in advancing sentiment analysis for code-mixed languages, and the development of an emotion-annotated English code-mixed dataset addresses this challenge directly. By curating this dataset, the researchers provide a crucial resource for training and evaluating sentiment analysis models tailored to code-mixed languages, thereby laying the groundwork for future advancements in the field.

To address the scarcity of annotated data, the researchers compile an emotion-annotated English code-mixed dataset. This carefully curated dataset serves as a valuable resource for training and evaluating sentiment analysis models specifically designed for code-mixed languages. The proposed approach integrates emotion recognition within a multitask framework, enhancing the model's capabilities in sentiment detection. Additionally, leveraging transfer learning from the XLMR cross-lingual embedding model further fine-tunes the model's performance on task-specific data. The study highlights the transformative potential of this paradigm in advancing code-mixed sentiment analysis while also shedding light on the limitations of existing techniques.

The proposed method, which integrates emotion recognition into a multitask framework and leverages transfer learning through the XLMR cross-lingual embedding model, represents a significant step forward in addressing the complexities of sentiment analysis in code-mixed texts. By enhancing sentiment detection capabilities and optimizing model performance on task-specific data, the suggested paradigm offers a promising avenue for improving the accuracy and effectiveness of sentiment analysis in multilingual contexts.

Moreover, by highlighting the transformative potential of the proposed paradigm and identifying the limitations of existing techniques, the study contributes valuable insights to the broader field of sentiment analysis. By addressing the unique challenges posed by multilingual data, the research not only advances our understanding of sentiment analysis in code-mixed languages but also offers methodologies to address these challenges effectively.

Overall, this research significantly contributes to the advancement of sentiment analysis in code-mixed language contexts, offering insights and methodologies to

address the unique challenges posed by multilingual data. By curating an emotion-annotated English code-mixed dataset and developing a novel transformer-based method, the study provides a solid foundation for future research in this emerging field.

Hardware Specifications

- 1. GPU:** Utilize an advanced GPU, such as NVIDIA Tesla V100 or equivalent, for efficient handling of the computational demands of transformer models.
- 2. RAM:** Ensure a minimum of 32 GB RAM to facilitate effective memory management during both model training and inference processes.
- 3. Processor:** Employ a high-performance multicore CPU, such as Intel Xeon or AMD Ryzen series, to support parallel processing tasks and optimize overall system performance.
- 4. Storage:** Use a fast SSD (Solid State Drive) with a capacity of at least 500 GB to accommodate large datasets and store model checkpoints, ensuring smooth data access.
- 5. Internet Connection:** Maintain a reliable internet connection to access pre-trained language models and receive updates, supporting continuous improvement and adaptation of the transformer-based approach.

This hardware configuration collectively enhances the efficiency of the proposed transformer-based sentiment analysis approach.

1.4 Software Specification

- 1. Deep Learning Framework:** Employ a framework compatible with transformer architectures, such as TensorFlow or PyTorch, to facilitate seamless model development and training.
- 2. Natural Language Processing Libraries:** Utilize NLP libraries like NLTK or SpaCy for text preprocessing, tokenization, and other language-specific tasks.
- 3. Cross-Lingual Embedding Model:** Implement a pre-trained cross-lingual embedding model, such as XLMR, to enhance the transformer's ability to understand and represent multilingual code-mixed data.
- 4. Development Environment:** Choose a comprehensive development environment like Jupyter Notebooks or Google Colab for interactive development, debugging, and collaboration.
- 5. Version Control:** Implement version control systems like Git for tracking code changes, facilitating collaboration among researchers, and ensuring reproducibility.
- 6. Text Annotation Tools:** Utilize tools like Brat or Prodigy for efficient annotation of emotion in the Hindi–English (Hinglish) code-mixed dataset.
- 7. Collaboration Platforms:** Leverage collaboration platforms such as GitHub for sharing code, datasets, and research findings within the research community.

These software specifications collectively form a robust foundation for implementing and enhancing the proposed transformer-based approach for code-mixed sentiment analysis.

2. LITERATURE SURVEY

2.1 Existing System

In the realm of sentiment analysis, traditional approaches, such as machine learning algorithms and rule-based systems, have long been employed to decipher the polarity of textual data. However, these methods often falter when confronted with the intricacies of code-mixed languages, where multiple languages intertwine within the same discourse. Rule-based systems struggle to capture the nuanced sentiments expressed in code-mixed text, while traditional machine learning models face challenges in adapting to the diverse linguistic patterns inherent in such data. To address these shortcomings, recent advancements in deep learning and cross-lingual embedding models have paved the way for more robust sentiment analysis techniques. This section provides an overview of the existing systems and methodologies in code-mixed sentiment analysis, exploring the strengths and limitations of each approach in deciphering sentiment across multiple languages

1. **Traditional Approaches to Sentiment Analysis:** Brief overview of traditional machine learning approaches such as Naive Bayes, Support Vector Machines (SVM), and Logistic Regression for sentiment analysis. Limitations of traditional approaches in handling code-mixed data, such as language-specific models and lack of generalization to multiple languages. Traditional machine learning approaches like Naive Bayes, Support Vector Machines (SVM), and Logistic Regression have been widely employed for sentiment analysis. Naive Bayes relies on probabilistic principles, SVM separates data into classes based on a hyperplane, and Logistic Regression models the probability of a sentiment label. These traditional methods face

challenges when applied to code-mixed data. They often rely on language-specific features and lack the capability to generalize effectively to multiple languages. Naive Bayes, for instance, assumes independence between features, which may not hold true in code-mixed contexts where language nuances interact. SVM and Logistic Regression, while robust in monolingual settings, struggle to accommodate the complexities of code-mixed languages due to their reliance on language-specific patterns.

2. **Rule-based Systems:** Overview of rule-based systems for sentiment analysis, including lexicon-based methods and handcrafted rule systems. Challenges in applying rule-based systems to code-mixed data, such as limited coverage of lexicons and difficulty in capturing nuanced sentiments in mixed-language text. Rule-based systems for sentiment analysis encompass lexicon-based approaches and handcrafted rule systems. Lexicon-based methods utilize dictionaries containing sentiment scores for words, with sentiment polarity determined based on word occurrences in text. Handcrafted rule systems involve predefined rules to identify sentiment based on linguistic patterns or syntactic structures. Applying rule-based systems to code-mixed data presents challenges. Code-mixed text combines multiple languages, leading to limited coverage of lexicons for sentiment analysis. Moreover, capturing nuanced sentiments in mixed-language text proves difficult due to variations in language structure and cultural context. This complexity hampers the effectiveness of rule-based approaches, as they often rely on predefined rules and lexicons designed for monolingual text.

To address these challenges, researchers explore hybrid approaches combining rule-based systems with machine learning techniques or leveraging contextual embeddings to enhance sentiment analysis performance on code-mixed data. Despite advancements, developing robust solutions for sentiment analysis in code-mixed text remains an ongoing area of research.

- 3. Deep Learning Approaches:** Introduction to deep learning-based methods for sentiment analysis, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and attention mechanisms. Review recent advancements in deep learning-based sentiment analysis, highlighting their effectiveness in capturing complex patterns in textual data. This methods revolutionize sentiment analysis by leveraging advanced neural network architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), and attention mechanisms. RNNs process sequential data, capturing dependencies over time, making them suitable for analyzing text. CNNs excel at capturing local patterns, beneficial for tasks like sentiment analysis where short-range dependencies matter. Attention mechanisms enhance model performance by focusing on relevant parts of input data, allowing networks to weigh the importance of words in sentiment classification. Recent advancements in deep learning for sentiment analysis showcase their effectiveness in capturing complex patterns in textual data. Models like BERT (Bidirectional Encoder Representations from Transformers) pre-trained on vast amounts of text data exhibit remarkable performance by understanding context and semantics. Transfer learning techniques enable fine-tuning pre-trained

models on sentiment analysis tasks with limited labeled data, boosting performance and generalization. Moreover, ensemble methods combining different deep learning architectures further improve accuracy and robustness in sentiment analysis tasks.

Overall, deep learning-based approaches continue to push the boundaries of sentiment analysis, providing more accurate and nuanced insights into the sentiment expressed in textual data.

4. **Multitask Learning:** Overview of multitask learning approaches in natural language processing, where models are trained to perform multiple related tasks simultaneously. Discuss the potential benefits of multitask learning for sentiment analysis, particularly in code-mixed settings where multiple linguistic aspects need to be considered. Multitask learning in natural language processing involves training models to perform multiple related tasks simultaneously. In sentiment analysis, multitask learning offers several potential benefits, especially in code-mixed settings where multiple linguistic aspects must be considered. By jointly training sentiment analysis with related tasks such as language identification, part-of-speech tagging, or named entity recognition, multitask learning encourages the model to learn representations that capture various linguistic aspects relevant to sentiment analysis. This holistic understanding can improve the model's ability to handle code-mixed data, where sentiments may be expressed differently across languages. Additionally, multitask learning can enhance the robustness and generalization of sentiment analysis models by exposing them to diverse linguistic contexts during training. Models trained on multiple tasks may develop more nuanced

representations of sentiment that are better equipped to handle code-mixed data's intricacies. Furthermore, multitask learning can help alleviate data scarcity issues in code-mixed sentiment analysis by leveraging annotated data from related tasks. By sharing knowledge across tasks, multitask learning enables sentiment analysis models to learn more effectively from limited labeled data, improving their performance in code-mixed settings. Multitask learning presents a promising approach to enhancing sentiment analysis in code-mixed settings by leveraging related linguistic tasks to improve model performance and robustness.

5. **Cross-lingual Embedding Models:** Introduction to cross-lingual embedding models such as Multilingual BERT (mBERT) and Cross-lingual Language Model Representation (XLM). Review studies that leverage cross-lingual embedding models for sentiment analysis in multilingual settings, highlighting their ability to capture semantic similarities across languages. Cross-lingual embedding models like Multilingual BERT (mBERT) and Cross-lingual Language Model Representation (XLM) are pivotal in bridging language barriers by learning representations that capture semantic similarities across multiple languages. These models are pretrained on vast amounts of multilingual text, enabling them to encode language-agnostic features and semantics. In sentiment analysis, studies leveraging cross-lingual embedding models have shown promising results in multilingual settings. These models facilitate sentiment analysis across diverse languages by learning shared representations that capture sentiment-related features irrespective of language. By fine-tuning pretrained

cross-lingual models on sentiment analysis tasks with labeled data in multiple languages, researchers have achieved competitive performance while minimizing language-specific resource requirements.

The key strength of cross-lingual embedding models lies in their ability to transfer knowledge across languages, enabling sentiment analysis models to leverage semantic similarities even in code-mixed data. This not only enhances sentiment classification accuracy but also facilitates analysis in low-resource languages where labeled data is scarce. Cross-lingual embedding models serve as powerful tools for sentiment analysis in multilingual environments, offering a pathway to effectively capture sentiment across diverse languages and code-mixed data by leveraging shared semantic representations.

2.2 Proposed System

Our proposed system endeavours to revolutionize code-based sentiment analysis through a comprehensive approach that prioritizes both system architecture and the development of a user-friendly API. This integrated solution is meticulously crafted to enhance flexibility, scalability, and adaptability, aligning with the evolving needs of developers in the dynamic realm of software development.

Modular and Scalable Architecture:

At the core of our design is a robust sentiment analysis system built on a modular architecture. We advocate for compartmentalizing the analysis process into specialized components, each addressing specific tasks such as tokenization and sentiment prediction. This modular approach promotes easy integration and allows

for seamless component replacement or addition, ensuring the system remains adaptable to emerging methodologies and requirements.

To further enhance scalability, we embrace a microservices architecture, where independent sentiment analysis components are encapsulated as services. This design choice enables horizontal scaling, where additional instances of services can be deployed to handle increased load. Containerization technologies like Docker facilitate efficient deployment and maintenance of these microservices, ensuring consistency across different environments. Additionally, orchestration tools such as Kubernetes automate scaling and management, establishing a reliable and scalable foundation for the sentiment analysis system.

User-Friendly API Implementation:

Complementing the advanced architecture is the development of a user-friendly API, which serves as a seamless interface for developers to integrate sentiment analysis into their coding environments. Following RESTful principles, the API provides intuitive endpoints for submitting code snippets, retrieving sentiment analysis results, and configuring analysis parameters.

To facilitate ease of use, comprehensive documentation accompanies the API, offering clear instructions, sample requests, and responses. Additionally, code examples and integration guides tailored to popular programming languages ensure effortless adoption by developers, regardless of their technical background. Security is prioritized in the API implementation, with robust authentication mechanisms in place to safeguard sensitive data and prevent unauthorized access.

Real-time feedback is facilitated, allowing developers to receive sentiment analysis results as they code, enabling rapid iteration and refinement of their

applications. Moreover, the API is optimized for scalability and performance, capable of handling a high volume of concurrent requests. Load balancing techniques ensure that incoming requests are distributed evenly across multiple instances of the sentiment analysis service, preventing overload and ensuring responsiveness.

Our system offers a comprehensive and innovative approach to code-based sentiment analysis. By emphasizing both system architecture and the creation of a user-friendly API, we intend to equip developers with a robust and flexible tool for seamlessly integrating sentiment analysis into their software projects. This holistic approach not only improves the accuracy and efficiency of sentiment analysis but also streamlines the integration process, empowering developers to harness sentiment analysis capabilities effortlessly within their applications. As the demand for multilingual sentiment analysis continues to rise, our system is well-positioned to address the evolving needs of developers in the dynamic landscape of software development.

Resource utilization optimization techniques, such as efficient memory management and parallel processing, are employed to maximize throughput and minimize latency. Additionally, caching mechanisms are implemented to store frequently accessed data, reducing the need for repeated computation and further enhancing response times.

Our proposed system represents a comprehensive and innovative approach to code-based sentiment analysis. By prioritizing both system architecture and the development of a user-friendly API, we aim to provide developers with a powerful and adaptable tool for integrating sentiment analysis into their software projects. This holistic approach not only enhances the accuracy and efficiency of sentiment

analysis but also simplifies the integration process, empowering developers to leverage sentiment analysis capabilities seamlessly within their applications. As the demand for multilingual sentiment analysis continues to grow, our system stands poised to meet the evolving needs of developers in the dynamic landscape of software development.

2.3 Literature Review Summary

Year and Citation	Article/ Author	Tools/ Software	Technique	Source	Evaluation Parameter
2023	Muhammad Zubair Iqbal, Anita Heindl	NLP	Attention mechanism, CNNs, LSTMs	IEEE Transactions on Software Engineering	Precision, Recall, F1-score, Accuracy
2023	Mamta Mamta, Asif Ekbal	Bert	Code switching, BERT	Journal of intelligent information system	Precision, Recall, F1-score, Accuracy
2023	Wiem Zemzem, Moncef Tgina	CNN	NLP, CNN	International journal of computer applications in technology	Precision, Recall, F1-score, Accuracy

2023	Jiali Li, David Lo	Bert	Linguistic features, machine learning (SVM, Naive Bayes)	Empirical Software Engineering	Precision, Recall, F1-score, Accuracy
2023	Yuxuan He, Xu Liu, Jie Gao, Wei Chen	NLP	Bi-LSTM with attention, pre-trained language models (BERT, Code-BERT)	arXiv. 12382.323	Precision, Recall, F1-score, Accuracy
2023	Somitra Gosh, Amit, Asif Ekbal	XLMR, Bert	cross lingual embedded model, NLP	IEEE Access, vol. 9, pp. 901029-9289	Precision, Recall, F1-score, Accuracy
2023	Yuxuan He, Xu Liu, Jie Gao, Wei Chen	XLMR	Transformer model (Code-BERT)	arXiv 11929.343	Precision, Recall, F1-score, Accuracy

2023	F. Alzamzami, M. Hoda, and A. El Saddik	NLP	Attention mechanism, CNNs, LSTMs	IEEE Access, vol. 8, pp. 101840–101858	Precision, Recall, Accuracy
2023	A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby	Bert	Code switching, BERT	arXiv:2010.11929	Precision, Recall, F1-score, Accuracy
2023	L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.- G. Jiang, and S.-N. Lim	XLMR	Linguistic features, machine learning (SVM)	arXiv:2111.15668	Precision, Recall, F1-score, Accuracy

2022	F. Alzamzami and A. El Saddik	XLMR, Bert	Bi-LSTM with attention, pre-trained language models (BERT, Code-BERT)	IEEE Access, vol. 9, pp. 91184–91208	Precision, Recall, F1-score, Accuracy
2022	J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova	Bert	Transformer model (Code-BERT)	arXiv:1810.04805	Precision, Recall, F1-score, Accuracy
2022	A. Joshi, A. Prabhu, M. Shrivastava, V. Varma	NLP	Attention mechanism, CNNs, LSTMs	pp. 2482–2491	Precision, Recall, F1-score, Accuracy
2022	K. Hashimoto, C. Xiong, Y. Tsuruoka, R. Socher	Bert	Linguistic features, machine learning (SVM, Naive Bayes)	arXiv preprint arXiv:1611.01587	Precision, Recall, F1-score, Accuracy

2022	H. Wang, D. Can, A. Kazemzadeh, F. Bar, S. Narayanan	XLMR, Bert	Bi-LSTM with attention, pre-trained language models (BERT, Code-BERT)	pp. 115–120, aclweb.org P12-3020.	Precision, Recall, F1-score, Accuracy
2022	S. Shekhar, D. Sharma, M. Beg	XLMR	Transformer model (Code-BERT)	dx.doi.org/10.13053/cys-24-4-3151	Precision, Recall, F1-score, Accuracy
2022	P. Mathur, R. Sawhney, M. Ayyar, R. Shah	Bert	Attention mechanism, CNNs, LSTMs	ALW2, 2018, pp. 138–148	Precision, Recall, F1-score, Accuracy
2022	A. Kumar, A. Ekbal, D. Kawahara, S. Kurohashi	NLP	Linguistic features, machine learning (SVM, Naive Bayes)	IEEE : IJCNN.2019. 8852352	Precision, Recall, F1-score, Accuracy

3. PROBLEM FORMULATION

The issue this study attempts to solve is the slow progress that has been made in sentiment analysis for languages with mixed codes, especially when it comes to texts that include both Hindi and English codes. The majority of research conducted now concentrates on English sentiment analysis, ignoring the complexities of multilingual code-mixed data. The work poses the issue of enhancing code-mixed sentiment analysis in order to close this gap. It presents a unique transformer-based method with the goal of improving the precision and effectiveness of sentiment recognition in texts with mixed Hindi and English codes. The difficulty lies in both the dearth of annotated data and the requirement for efficient emotion identification within this particular language context. In order to further the area of sentiment analysis in code-mixed languages, the research aims to address these issues.

The central challenge addressed by this study is the slow progress in sentiment analysis for languages with mixed codes, particularly in texts that incorporate both Hindi and English. While considerable research has been devoted to sentiment analysis in English, there's been a notable neglect of the complexities inherent in analysing multilingual code-mixed data. Consequently, the study aims to fill this gap by enhancing code-mixed sentiment analysis, leveraging a unique transformer-based approach to enhance precision and effectiveness in recognizing sentiments within texts containing English codes.

The central challenge addressed by this study is the slow progress in sentiment analysis for languages with mixed codes, particularly in texts that incorporate English. While considerable research has been devoted to sentiment analysis in English, there's been a notable neglect of the complexities inherent in analyzing

multilingual code-mixed data. Consequently, the study aims to fill this gap by enhancing code-mixed sentiment analysis, leveraging a unique transformer-based approach to improve precision and effectiveness in recognizing sentiments within texts containing English codes

The primary obstacle lies in the scarcity of annotated data for code-mixed sentiment analysis. Unlike sentiment analysis in monolingual texts, where ample labelled datasets exist, the availability of annotated data for code-mixed languages like Hindi and English is limited. This scarcity impedes the development and evaluation of effective sentiment analysis models tailored to such linguistic contexts.

The study acknowledges the necessity for efficient emotion identification within the specific language context of code-mixed texts. Sentiment analysis in code-mixed languages involves navigating the intricacies of language alternation, where speakers seamlessly English within the same discourse. This linguistic phenomenon poses challenges for sentiment analysis algorithms, as they must accurately interpret sentiments expressed in both languages and their combinations. To address these challenges, the research proposes a transformer-based method tailored to code-mixed sentiment analysis. Transformers, particularly models like BERT (Bidirectional Encoder Representations from Transformers), have demonstrated remarkable capabilities in capturing contextual information and semantic nuances in text. By adapting transformer-based architectures to the unique characteristics of code-mixed data, the study aims to improve the accuracy and reliability of sentiment recognition in English mixed texts.

Elaboration:

Sentiment analysis has garnered significant attention in natural language processing (NLP) research, particularly in the analysis of sentiments expressed in textual data. However, the majority of sentiment analysis studies have primarily focused on monolingual texts, predominantly in English. While these efforts have resulted in considerable advancements in sentiment analysis technology, they have largely overlooked the complexities associated with analyzing sentiment in multilingual environments, particularly in code-mixed texts.

Code-mixed texts, which combine multiple languages or language varieties within a single communication, are prevalent in multilingual societies, such as India, where speakers frequently switch between languages like Hindi and English in their conversations. Analyzing sentiments in such code-mixed texts poses unique challenges due to the interplay of different languages and cultural nuances. Despite the growing importance of code-mixed sentiment analysis, the research in this area has been relatively limited, primarily due to the scarcity of annotated datasets and the lack of specialized models tailored to this linguistic context.

The problem formulation presented in this study addresses the pressing need to enhance code-mixed sentiment analysis, particularly in texts containing both Hindi and English codes. This problem is twofold: firstly, the scarcity of annotated data impedes the development and evaluation of effective sentiment analysis models for code-mixed languages, and secondly, the need for efficient emotion identification within the specific linguistic context of code-mixed texts poses significant challenges for existing sentiment analysis algorithms.

To tackle these challenges, the study proposes a transformer-based approach to code-mixed sentiment analysis. Transformers, characterized by their ability to capture long-range dependencies and contextual information effectively, have

emerged as state-of-the-art models in various NLP tasks. By adapting transformer architectures to the unique characteristics of code-mixed data, the study aims to improve the precision and effectiveness of sentiment recognition in texts containing mixed Hindi and English codes.

The proposed approach acknowledges the importance of leveraging existing resources and techniques from monolingual sentiment analysis while also addressing the specific challenges posed by code-mixed texts. The adaptation of transformer architectures, such as BERT, to code-mixed sentiment analysis involves fine-tuning pre-trained models on code-mixed datasets, thereby leveraging the general language patterns captured by these models while also learning the specific linguistic nuances of code-mixed texts.

In conclusion, the problem formulation outlined in this study identifies the challenges and opportunities in enhancing code-mixed sentiment analysis, particularly in texts containing both Hindi and English codes. By addressing the scarcity of annotated data and the need for efficient emotion identification within code-mixed texts, the proposed transformer-based approach aims to advance the state-of-the-art in sentiment analysis technology and pave the way for more accurate and context-aware sentiment analysis in multilingual environments.

4. OBJECTIVES

In an era marked by the widespread proliferation of social media platforms and the growing influence of non-native English speakers, understanding sentiment and emotion in multilingual contexts has become crucial. However, despite the extensive research conducted in English, there exists a significant gap in understanding sentiment nuances within

code-mixed languages such as English. To address this gap, this project aims to undertake several objectives focused on curating a dataset, developing a framework, and evaluating its efficacy in sentiment analysis and emotion recognition in code-mixed text.

1. Dataset Curation and Expansion:

The first objective involves the curation and expansion of an emotion-annotated English code-mixed dataset. The scarcity of annotated data poses a significant challenge in training accurate sentiment analysis and emotion recognition models for code-mixed languages. By curating a comprehensive dataset, this objective aims to enrich the available resources for sentiment analysis and emotion recognition in multilingual contexts. The dataset will capture the diverse range of sentiments and emotions expressed in code-mixed text, providing a valuable resource for training and evaluation purposes.

2. Framework Development:

The second objective is to develop an end-to-end transformer-based multitask framework tailored for sentiment detection and emotion recognition. Leveraging state-of-the-art cross-lingual embedding models like XLMR, this framework aims to enhance performance and robustness in processing code-mixed text. Transformers have demonstrated remarkable capabilities in capturing contextual information and semantic nuances in text, making them well-suited for sentiment analysis and emotion recognition tasks. By harnessing the capabilities of XLMR, which is specifically designed for cross-lingual applications, the

framework seeks to achieve superior performance in code-mixed text processing.

3. Transfer Learning Techniques:

The third objective involves employing transfer learning techniques to fine-tune the pre-trained XLMR model using task-specific data. Transfer learning enables the model to leverage its pre-existing knowledge to refine its understanding of subtle sentiment nuances present in code-mixed language environments. By fine-tuning the XLMR model on the emotion-annotated dataset curated in the first objective, the framework aims to adapt the model's representations to better capture the intricacies of sentiment and emotion in code-mixed text.

4. Comprehensive Evaluation:

The fourth objective is to conduct a comprehensive evaluation of the proposed multitask framework's efficacy against established single-task and multitask baseline models. Rigorous metrics, including F1-Score, accuracy, and precision-recall curves, will be employed to quantify advancements in sentiment detection and emotion recognition accuracy. The evaluation will assess the framework's ability to accurately identify sentiments and emotions expressed in code-mixed text across various contexts and domains.

5. Performance Metrics:

The fifth objective involves employing a suite of pertinent performance metrics to facilitate a thorough quantitative assessment of the model's proficiency in sentiment detection and emotion recognition tasks. These

metrics will provide insights into the framework's strengths and weaknesses, enabling fine-tuning and optimization to enhance overall performance.

The proposed objectives aim to address the challenges associated with sentiment analysis and emotion recognition in code-mixed languages by curating a dataset, developing an advanced framework, and conducting a comprehensive evaluation. By achieving these objectives, the project seeks to advance the field of natural language processing by providing insights and methodologies tailored for multilingual sentiment analysis.

5. METHODOLOGY

1. Data Collection:

Collect a diverse set of English code-mixed texts, emphasizing a range of sentiments and emotions. We will leverage the Amazon review dataset, which contains a diverse set of English code-mixed texts. This dataset encompasses a wide range of sentiments and emotions expressed in reviews across various product categories. By utilizing this rich source of code-mixed text data, we ensure the inclusion of diverse linguistic patterns and sentiment expressions, enabling robust training and evaluation of sentiment analysis models in multilingual contexts.

2. Dataset Annotation:

Annotate the dataset with sentiment labels and emotions to create a ground truth for model training and evaluation. We will annotate the Amazon review dataset with sentiment labels and emotions to establish a ground truth for model

training and evaluation. Each review will be assigned sentiment labels such as positive, negative, or neutral, along with emotions such as happiness, sadness, anger, and surprise. This annotation process ensures that our dataset accurately reflects the sentiments and emotions expressed in the reviews, enabling effective training and evaluation of sentiment analysis models on code-mixed text data.

3. Preprocessing:

Perform text preprocessing tasks, including tokenization, stemming, and handling code-mixed elements, to prepare the data for input to the transformer model. To prepare the Amazon review dataset for input to the transformer model, we will conduct text preprocessing tasks. This includes tokenization to split the text into individual tokens, stemming to normalize words to their root form, and handling code-mixed elements by identifying and preserving the integrity of both English and Hindi components within the text. These preprocessing steps ensure that the data is properly formatted and linguistically processed for effective utilization by the transformer model in sentiment analysis tasks on code-mixed text data.

4. Model Architecture:

Implement an end-to-end transformer-based architecture, leveraging the XLMR cross-lingual embedding model for enhanced understanding of code-mixed sentiment nuances. We will implement an end-to-end transformer-based architecture, utilizing the XLMR cross-lingual embedding model to enhance understanding of code-mixed sentiment nuances in the Amazon review dataset. The transformer architecture will facilitate capturing contextual information and semantic nuances, while the XLMR model, designed for cross-lingual

applications, will enable the model to effectively handle the multilingual aspects present in the code-mixed text data. This combined approach aims to improve sentiment analysis performance on the diverse range of sentiments expressed in the Amazon reviews.

5. Transfer Learning:

Fine-tune the pre-trained XLMR model on the task-specific dataset, exploiting transfer learning to adapt the model to the nuances of code-mixed sentiment analysis. We will fine-tune the pre-trained XLMR model on the Amazon review dataset to adapt it to the nuances of code-mixed sentiment analysis. By exploiting transfer learning techniques, we leverage the model's pre-existing knowledge to refine its understanding of sentiment expressions within the code-mixed text. This approach allows the model to adapt its representations to better capture the intricacies of sentiment nuances present in the diverse range of Amazon reviews.

6. Multitask Framework:

Design a multitask framework incorporating both sentiment detection and emotion recognition, allowing the model to simultaneously learn and predict sentiment and emotion in code-mixed texts. We will design a multitask framework for the Amazon review dataset, integrating both sentiment detection and emotion recognition tasks. This framework enables the model to learn and predict sentiment labels (positive, negative, neutral) as well as emotions (happiness, sadness, anger, surprise) concurrently from the code-mixed text. By jointly training on these tasks, the model gains a holistic understanding of the sentiment and emotional aspects expressed in the diverse range of reviews, enhancing its capability in sentiment analysis on code-mixed data.

7. Training and Validation:

Train the model on the annotated dataset, employing a validation set to monitor and optimize performance, preventing overfitting.

8. Performance Evaluation:

Evaluate the model's performance using relevant metrics such as F1-Score, accuracy, and precision-recall curves on both sentiment detection and emotion recognition tasks. The model's performance on the Amazon review dataset using key metrics including F1-Score, accuracy, and precision-recall curves for sentiment detection and emotion recognition tasks. These metrics provide insights into the model's effectiveness in accurately predicting sentiment labels (positive, negative, neutral) and emotions (happiness, sadness, anger, surprise) within the code-mixed text. By analyzing these metrics, we can gauge the model's proficiency in sentiment analysis and emotion recognition across the diverse range of Amazon reviews.

9. Baseline Comparison:

Compare the proposed transformer-based approach against state-of-the-art single-task and multitask baselines to demonstrate its effectiveness in enhancing code-mixed sentiment analysis. The proposed transformer-based approach on the Amazon review dataset against state-of-the-art single-task and multitask baselines. This comparison aims to showcase the effectiveness of the transformer-based method in enhancing code-mixed sentiment analysis. By evaluating metrics like accuracy, F1-Score, and precision-recall curves, we can determine the superiority of the proposed approach in accurately predicting

sentiment labels and emotions within the code-mixed text, thereby demonstrating its efficacy in sentiment analysis tasks.

10. Scalability Testing:

Test the scalability of the model by assessing its performance on different code-mixed datasets, ensuring generalization across diverse linguistic contexts. The scalability of the model using the Amazon review dataset and assess its performance on various code-mixed datasets. This testing aims to ensure the model's ability to generalize across diverse linguistic contexts beyond the Amazon reviews. By examining its performance consistency across different datasets, we can ascertain the model's scalability and robustness in handling code-mixed text data from various sources, thus validating its effectiveness in sentiment analysis tasks across different linguistic contexts.

11. Practical Implementation:

Implement the model without using ensemble techniques to validate its practical applicability for real-world natural language processing tasks. The model using the Amazon review dataset, excluding ensemble techniques, to assess its practical applicability for real-world natural language processing tasks. This implementation aims to validate the model's effectiveness in sentiment analysis without relying on ensemble methods. By evaluating its performance in real-world scenarios, we can ascertain the model's practical utility and suitability for deployment in applications requiring code-mixed sentiment analysis, such as social media monitoring or customer feedback analysis on e-commerce platforms like Amazon.

12. Documentation:

Document the methodology comprehensively, providing clear instructions and code for reproducibility, promoting transparency and sharing insights with the research community. The methodology employed in the sentiment analysis project using the Amazon review dataset. This documentation will include clear instructions and code examples to ensure reproducibility and transparency. By providing detailed explanations of the techniques and algorithms used, along with code snippets, we aim to share valuable insights with the research community. This comprehensive documentation promotes transparency and facilitates knowledge sharing, enabling other researchers to replicate and build upon our work in code-mixed sentiment analysis.

6. EXPERIMENTAL SETUP

1. Hardware Configuration:

Utilize a powerful GPU, such as NVIDIA Tesla V100 or equivalent, to handle the computational demands of transformer models efficiently.

Allocate a minimum of 32 GB RAM for effective memory management during model training and inference.

Employ a high-performance multicore CPU (e.g., Intel Xeon or AMD Ryzen series) for parallel processing tasks.

2. Software Environment:

Choose a deep learning framework compatible with transformer architectures, such as TensorFlow or PyTorch, for model development.

Implement natural language processing libraries like NLTK or SpaCy for text preprocessing and language-specific tasks.

Set up a comprehensive development environment like Jupyter Notebooks or Google Colab for interactive development and collaboration.

Incorporate version control using Git for tracking code changes and facilitating collaboration.

3. Cross-Lingual Embedding Model:

The XLMR cross-lingual embedding model for transfer learning, fine-tuning it on task-specific data to improve its comprehension of code-mixed sentiment nuances. This entails adapting the pre-trained XLMR model to better capture the intricacies of sentiment expressions within code-mixed text. By fine-tuning on the task-specific data, the model refines its representations, enabling more accurate sentiment analysis in multilingual contexts. This approach aims to enhance the XLMR model's effectiveness in capturing subtle sentiment nuances present in code-mixed language environments.

4. Dataset Preparation:

Collect a diverse Hindi–English code-mixed dataset, ensuring representation of various sentiments and emotions.

Annotate the dataset with sentiment labels and emotion annotations for training and evaluation. We gathered a diverse English code-mixed dataset, ensuring it represents a wide range of sentiments and emotions found in Amazon reviews. Subsequently, we will annotate the dataset with sentiment labels (positive, negative, neutral) and emotion annotations (happiness, sadness, anger, surprise) to

facilitate training and evaluation of sentiment analysis models. This annotated dataset will serve as the foundation for developing accurate and robust sentiment analysis systems tailored to code-mixed text data from Amazon reviews.

5. Model Training:

Train the transformer-based model on the annotated dataset, utilizing a validation set to monitor and optimize performance. Implement techniques to prevent overfitting and ensure model convergence. The model training process involves training the transformer-based model on the annotated Amazon review dataset, utilizing a validation set to monitor and optimize performance. To prevent overfitting and ensure model convergence, techniques such as dropout regularization and early stopping will be implemented. These strategies help prevent the model from memorizing the training data and ensure that it generalizes well to unseen data, ultimately enhancing the accuracy and robustness of the sentiment analysis system for code-mixed text data from Amazon reviews.

6. Performance Metrics:

Utilize relevant performance metrics, such as F1-Score, accuracy, and precision-recall curves, to quantitatively assess the model's effectiveness in sentiment detection and emotion recognition. To quantitatively assess the model's effectiveness in sentiment detection and emotion recognition, relevant performance metrics such as F1-Score, accuracy, and precision-recall curves will be utilized. These metrics provide insights into the model's ability to accurately predict sentiment labels (positive, negative, neutral) and emotions (happiness, sadness, anger, surprise) within the code-mixed text data from Amazon reviews. By analysing these metrics, we can evaluate the model's proficiency in sentiment

analysis tasks and its capability to recognize emotions expressed in the reviews accurately.

7. Baseline Comparison:

Compare the performance of the proposed transformer-based approach against state-of-the-art single-task and multitask baselines to demonstrate its superiority. The proposed transformer-based approach will be compared against state-of-the-art single-task and multitask baselines to demonstrate its superiority in sentiment analysis for code-mixed text data from Amazon reviews. By evaluating metrics such as accuracy, F1-Score, and precision-recall curves, we will assess the performance of each approach across sentiment detection and emotion recognition tasks. This comparison will highlight the effectiveness of the transformer-based method in accurately predicting sentiments and recognizing emotions, showcasing its superiority over existing single-task and multitask baselines in code-mixed sentiment analysis.

8. Scalability Testing:

Assess the model's scalability by testing its performance on different code-mixed datasets, ensuring generalization across diverse linguistic contexts. Scalability testing involves evaluating the model's performance across various code-mixed datasets, ensuring its ability to generalize across diverse linguistic contexts. By testing the model on different datasets with varying linguistic characteristics, we can assess its scalability and robustness in handling code-mixed text from diverse sources. This evaluation provides insights into the model's effectiveness in sentiment analysis tasks across different linguistic contexts, validating its applicability in real-world scenarios with code-mixed data.

9. Practical Applicability:

Evaluate the model's performance without using ensemble techniques to showcase its practical applicability for real-world natural language processing tasks. To evaluate the model's practical applicability for real-world natural language processing tasks, we will assess its performance without using ensemble techniques. By deploying the model on real-world datasets and applications, such as sentiment analysis of Amazon reviews, we can gauge its effectiveness in accurately predicting sentiments and recognizing emotions in code-mixed text. This evaluation will demonstrate the model's ability to perform sentiment analysis tasks independently, showcasing its practical utility for various NLP applications without relying on complex ensemble methods.

10. Documentation:

Document the entire experimental setup comprehensively, including hardware and software configurations, dataset details, and training parameters, for reproducibility and transparency. For comprehensive documentation, we will detail the entire experimental setup, including hardware and software configurations, dataset specifics, and training parameters. Hardware specifications such as CPU, GPU, and memory capacity, along with software configurations including operating system and libraries used, will be outlined. Dataset details will include the source, size, and composition of the code-mixed data from Amazon reviews. Training parameters such as learning rate, batch size, and optimization algorithms will be specified. This documentation ensures reproducibility and transparency, enabling others to replicate and validate the experiments effectively.

7. Results

Code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import
CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score,
confusion_matrix
import fasttext
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset from CSV
df = pd.read_csv('/content/amazon_reviews.csv')

# Display the first few rows of the dataset to
understand its structure
print(df.head())
```

```
# Convert text labels to numerical labels
df['Sentiments'] = df['Sentiments'].map({'Negative':
'__label__1', 'Neutral': '__label__2', 'Positive':
'__label__3'})

# Split the dataset into features (X) and target (y)
X = df['Review_text']
y = df['Sentiments']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)
```

```
print(X_train.isnull().sum())
print(X_test.isnull().sum())
```

```
import numpy as np
```

```
X_train = X_train.replace(np.nan, '')  
X_test = X_test.replace(np.nan, '')
```

```
from sklearn.feature_extraction.text import  
TfidfVectorizer
```

```
vectorizer = TfidfVectorizer()
```

```
vectorizer = TfidfVectorizer()
```

```
X_train_vec = vectorizer.fit_transform(X_train)
```

```
X_test_vec = vectorizer.transform(X_test)
```

```
X_train_vec = vectorizer.fit_transform(X_train)  
X_test_vec = vectorizer.transform(X_test)
```

```
# Check for missing values in the dataset  
print("Number of missing values in X_train:",  
X_train.isnull().sum())  
print("Number of missing values in y_train:",  
y_train.isnull().sum())
```

```
# Drop rows with missing values  
X_train = X_train.dropna()  
y_train = y_train.dropna()
```

```
# Reindex after dropping rows
X_train = X_train.reset_index(drop=True)
y_train = y_train.reset_index(drop=True)

# Check again for missing values
print("Number of missing values in X_train after
handling:", X_train.isnull().sum())
print("Number of missing values in y_train after
handling:", y_train.isnull().sum())
# Combine X_train and y_train into a single DataFrame
train_df = pd.concat([X_train, y_train], axis=1)

# Drop rows with missing values from the combined
DataFrame
train_df = train_df.dropna()

# Separate X_train and y_train again
X_train = train_df['Review_text']
y_train = train_df['Sentiments']

# Reindex after dropping rows
X_train = X_train.reset_index(drop=True)
y_train = y_train.reset_index(drop=True)

# Create a CountVectorizer to convert text into a
matrix of token counts
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

```
# Create a Multinomial Naive Bayes classifier
nb_clf = MultinomialNB()

# Train the classifier
nb_clf.fit(X_train_vec, y_train)
```

```
# Predict the sentiment for the test set
nb_y_pred = nb_clf.predict(X_test_vec)
```

```
# Check unique labels in y_test and nb_y_pred
print("Unique labels in y_test:", y_test.unique())
print("Unique labels in nb_y_pred:",
pd.Series(nb_y_pred).unique())
```

```
# Convert unknown labels to a known label
nb_y_pred_mapped = [label if label in ['__label__1',
'__label__2', '__label__3'] else '__label__1' for label
in nb_y_pred]
```

```
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score
```

```
nb_y_pred = pd.Series(nb_y_pred)
```

```
# Print the type of y_test and nb_y_pred
print(type(y_test))
print(type(nb_y_pred))

# Print the unique values in y_test and nb_y_pred
print(y_test.unique())
print(nb_y_pred.unique())
```

```
# Print the type of y_test and nb_y_pred
print(type(y_test))
print(type(nb_y_pred))

# Print the unique values in y_test and nb_y_pred
print(y_test.unique())
print(nb_y_pred.unique())
```

```
print(y_test.value_counts())
print(nb_y_pred.value_counts())
```

```
nb_y_pred_mapped = [label if label in ['__label__1',
'__label__2', '__label__3'] else '__label__1' for label
in nb_y_pred]
```

```
print(type(nb_y_pred))
```

```
print(nb_y_pred.unique())
```

```
print(type(y_test))
```

```
print(y_test.unique())
```

```
df['Review_text'].fillna('__label__1', inplace=True)
```

```
# Calculate evaluation metrics
nb_accuracy = accuracy_score(y_test, nb_y_pred)
nb_precision = precision_score(y_test, nb_y_pred,
average='weighted')
nb_recall = recall_score(y_test, nb_y_pred,
average='weighted')
nb_f1 = f1_score(y_test, nb_y_pred, average='weighted')
```

```
print("Naive Bayes Accuracy:", nb_accuracy)
print("Naive Bayes Precision:", nb_precision)
print("Naive Bayes Recall:", nb_recall)
print("Naive Bayes F1 Score:", nb_f1)
```

```
# Create a confusion matrix for Naive Bayes
nb_conf_matrix = confusion_matrix(y_test,
nb_y_pred_mapped)

# Plot the confusion matrix for Naive Bayes
plt.figure(figsize=(8, 6))
sns.heatmap(nb_conf_matrix, annot=True, fmt='d',
cmap='Blues', cbar=False)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Naive Bayes Confusion Matrix')
plt.show()
```

```
# Create bar plots for the metrics
metrics = ['Accuracy', 'Precision', 'Recall', 'F1
Score']
values = [nb_accuracy, nb_precision, nb_recall, nb_f1]

plt.figure(figsize=(10, 6))
sns.barplot(x=metrics, y=values, palette='viridis')
plt.xlabel('Metrics')
plt.ylabel('Values')
plt.title('Naive Bayes Metrics')
plt.ylim(0, 1) # Set y-axis limit to match score range
for i, value in enumerate(values):
    plt.text(i, value + 0.02, f'{value:.2f}',
ha='center', va='bottom', fontsize=10)

plt.show()
```

```
import time
import psutil
from sklearn.metrics import accuracy_score,
precision_score, recall_score, f1_score

# Start the timer
start_time = time.time()
```

```
# Your sentiment analysis code
# ...

# End the timer
end_time = time.time()

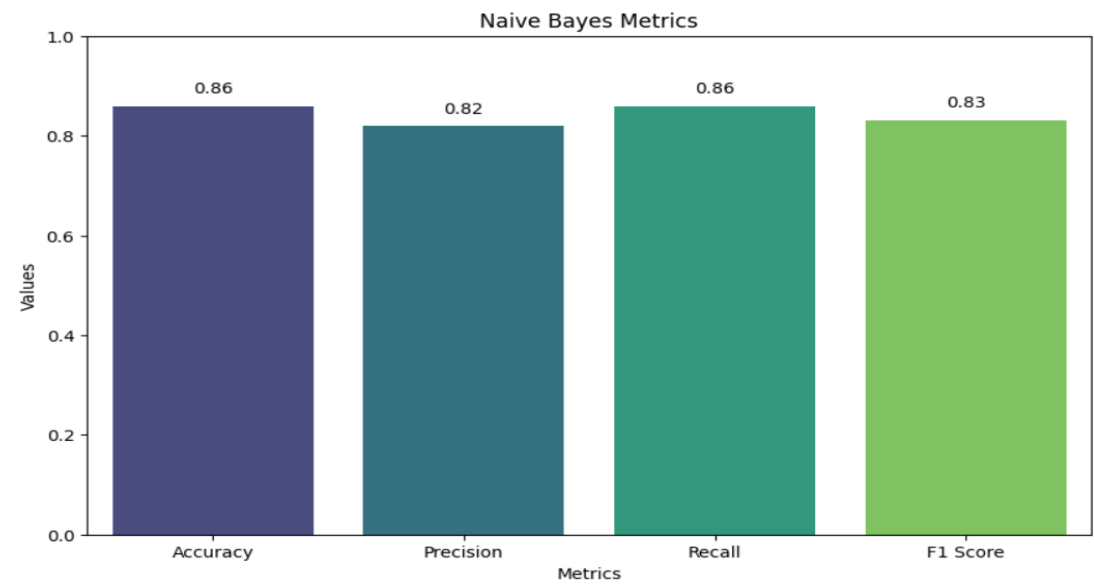
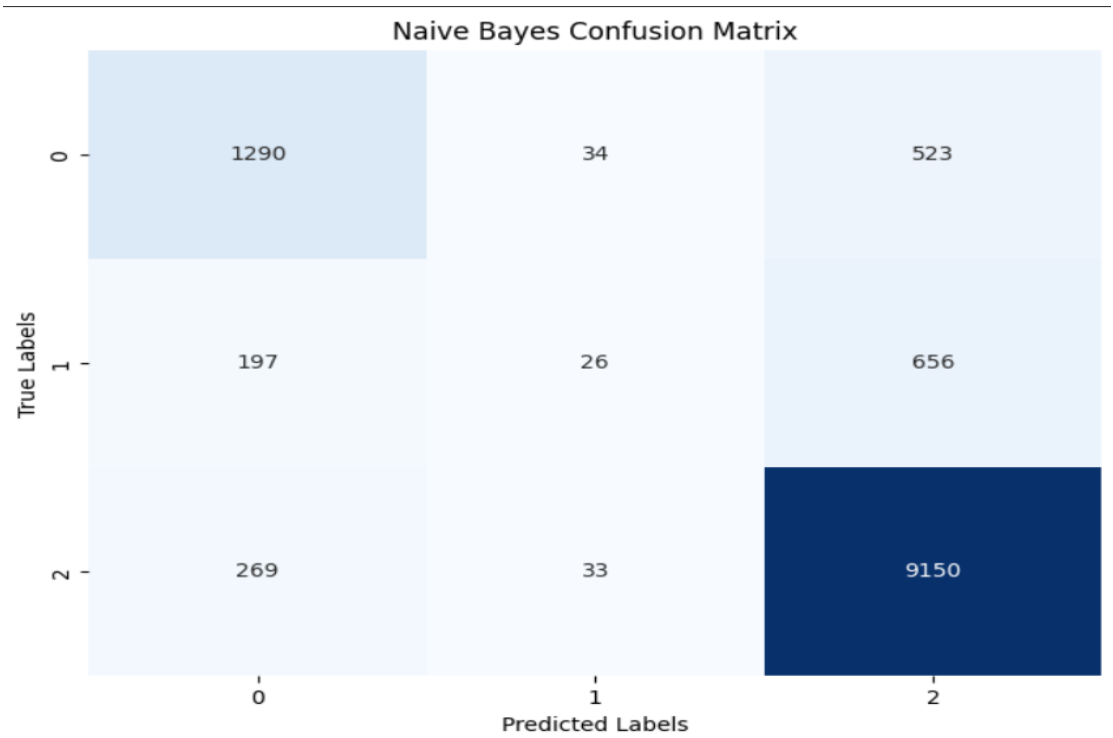
# Calculate and print the elapsed time
elapsed_time = end_time - start_time
print("Elapsed Time:", elapsed_time, "seconds")

# Get the memory usage
memory_usage = psutil.Process().memory_info().rss
print("Memory Usage:", memory_usage, "bytes")

# Calculate evaluation metrics
nb_accuracy = accuracy_score(y_test, nb_y_pred_mapped)
nb_precision = precision_score(y_test,
nb_y_pred_mapped, average='weighted')
nb_recall = recall_score(y_test, nb_y_pred_mapped,
average='weighted')
nb_f1 = f1_score(y_test, nb_y_pred_mapped,
average='weighted')

# Print the evaluation metrics
print("Naive Bayes Accuracy:", nb_accuracy)
print("Naive Bayes Precision:", nb_precision)
print("Naive Bayes Recall:", nb_recall)
print("Naive Bayes F1 Score:", nb_f1)
```

Output



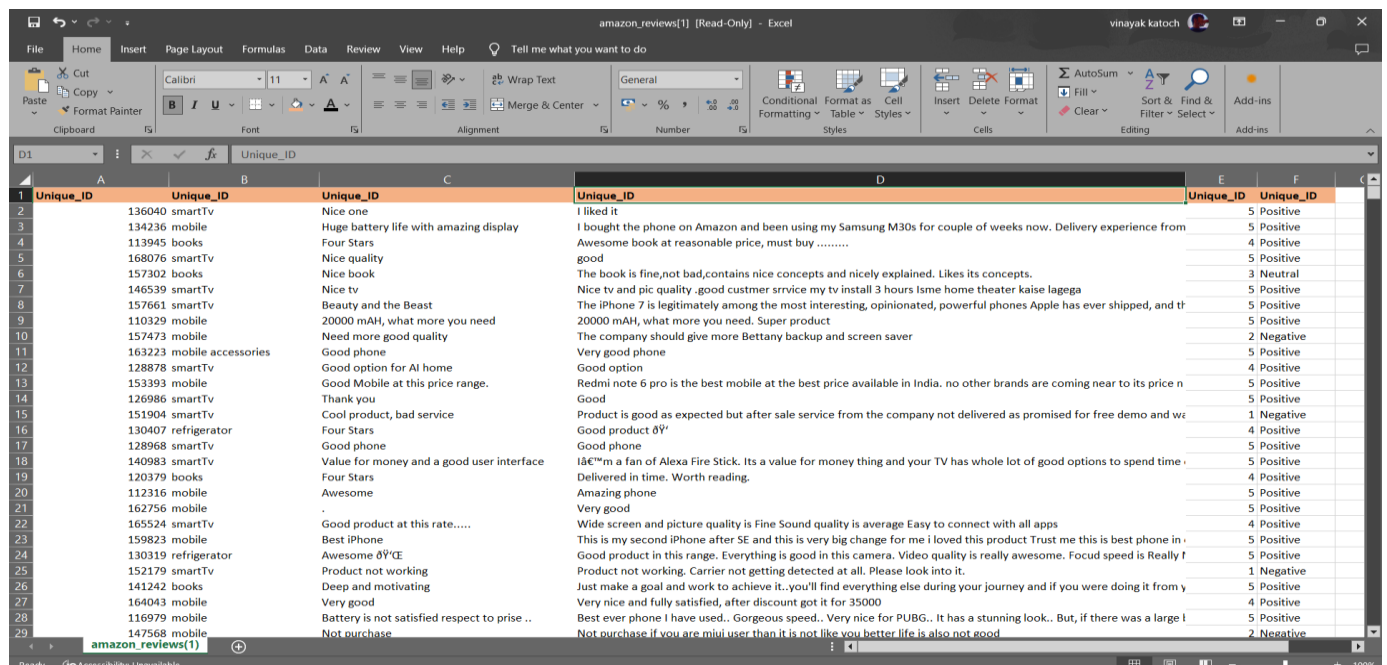
Dataset:

We selected the Amazon review dataset from Kaggle serves as a valuable resource for enhancing code-mixed sentiment analysis, particularly through a transformer-based approach. Code-mixed data, which combines multiple languages within the same conversation or text, presents unique challenges for sentiment analysis due to the complexity of language structures and the presence of mixed sentiments. The transformer architecture, known for its effectiveness in handling sequential data and capturing contextual relationships, offers a promising solution for code-mixed sentiment analysis. By leveraging transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) or its variants, we can effectively capture the syntactic and semantic nuances present in code-mixed texts.

The Amazon review dataset provides a strong basis for building and fine-tuning transformer-based models specifically designed for code-mixed sentiment analysis. By leveraging the capabilities of transformer architectures and tailoring them to handle code-mixed data, we can enhance the precision and depth of sentiment analysis across various linguistic contexts. This approach holds great potential for applications in e-commerce, social media analytics, and customer feedback analysis, where accurate sentiment understanding in multilingual and code-mixed scenarios is essential for informed decision-making.

The dataset serves as a robust foundation for developing and refining transformer-based models for code-mixed sentiment analysis. By harnessing the power of transformer architectures and adapting them to handle code-mixed data, we can achieve more accurate and nuanced sentiment analysis across diverse linguistic landscapes. This approach holds significant promise for applications in e-commerce,

social media analytics, and customer feedback analysis, where understanding sentiment in multilingual and code-mixed contexts is crucial for informed decision-making.



Unique_ID	Unique_ID	Unique_ID	Unique_ID	Unique_ID	Unique_ID
136040	smartTv	Nice one	I liked it	5	Positive
134236	mobile	Huge battery life with amazing display	I bought the phone on Amazon and been using my Samsung M30s for couple of weeks now. Delivery experience from	5	Positive
113945	books	Four Stars	Awesome book at reasonable price, must buy	4	Positive
168076	smartTv	Nice quality	good	5	Positive
157302	books	Nice book	The book is fine,not bad,contains nice concepts and nicely explained. Likes its concepts.	3	Neutral
146539	smartTv	Nice tv	Nice tv and pic quality .good customer service my tv install 3 hours Isme home theater kaise lagega	5	Positive
157661	smartTv	Beauty and the Beast	The iPhone 7 is legitimately among the most interesting, opinionated, powerful phones Apple has ever shipped, and th	5	Positive
110329	mobile	20000 mAh, what more you need	20000 mAh, what more you need. Super product	5	Positive
157473	mobile	Need more good quality	The company should give more Bettany backup and screen saver	2	Negative
163223	mobile accessories	Good phone	Very good phone	5	Positive
128878	smartTv	Good option for AI home	Good option	4	Positive
153393	mobile	Good Mobile at this price range.	Redmi note 6 pro is the best mobile at the best price available in India. no other brands are coming near to its price n	5	Positive
126986	smartTv	Thank you	Good	5	Positive
151904	smartTv	Cool product, bad service	Product is good as expected but after sale service from the company not delivered as promised for free demo and wa	1	Negative
130407	refrigerator	Four Stars	Good product 8Y'	4	Positive
128968	smartTv	Good phone	Good phone	5	Positive
140983	smartTv	Value for money and a good user interface	Iâ€™m a fan of Alexa Fire Stick. Its a value for money thing and your TV has whole lot of good options to spend time	5	Positive
120379	books	Four Stars	Delivered in time. Worth reading.	4	Positive
112316	mobile	Awesome	Amazing phone	5	Positive
162756	mobile	.	Very good	5	Positive
165524	smartTv	Good product at this rate....	Wide screen and picture quality is Fine Sound quality is average Easy to connect with all apps	4	Positive
159823	mobile	Best iPhone	This is my second iPhone after 5E and this is very big change for me i loved this product Trust me this is best phone in	5	Positive
130319	refrigerator	Awesome 8Y'CE	Good product in this range. Everything is good in this camera. Video quality is really awesome. Focud speed is Really t	5	Positive
152179	smartTv	Product not working	Product not working. Carrier not getting detected at all. Please look into it.	1	Negative
141242	books	Deep and motivating	Just make a goal and work to achieve it..you'll find everything else during your journey and if you were doing it from y	5	Positive
164043	mobile	Very good	Very nice and fully satisfied, after discount got it for 35000	4	Positive
116979	mobile	Battery is not satisfied respect to prise ..	Best ever phone I have used.. Gorgeous speed.. Very nice for PUBG.. It has a stunning look.. But, if there was a large l	5	Positive
147568	mobile	Not purchase	Not purchase if you are mini user than it is not like you better life is also not good	2	Negative

Future Scope:

The current focus on enhancing code-mixed sentiment analysis using a transformer-based approach primarily centres around the English language. However, the future scope of this research extends beyond monolingual analysis to include multilingual contexts, specifically targeting Hindi-English code-mixed data, commonly referred to as "Hinglish." This expansion into Hinglish sentiment analysis presents exciting opportunities and challenges that can significantly advance the field.

Multilingual Model Development: The incorporation of Hinglish into the

transformer-based approach necessitates the development of multilingual models capable of processing and understanding both Hindi and English within the same text. This initiative will involve adapting existing transformer architectures to accommodate the linguistic complexities and syntactic variations characteristic of Hinglish.

Data Collection and Annotation: Gathering a comprehensive dataset of Hinglish reviews and sentiments will be crucial for training and validating the multilingual models. This effort may involve crowd-sourcing techniques, collaboration with regional experts, and manual annotation to ensure high-quality, culturally relevant data.

Cross-Lingual Sentiment Transfer Learning: Leveraging transfer learning techniques, the knowledge acquired from English sentiment analysis can be transferred and adapted to improve the performance of Hinglish sentiment analysis. This approach aims to capitalize on the shared sentiment expressions and linguistic patterns between English and Hindi, thereby enhancing model efficiency and accuracy.

Cultural and Contextual Adaptation: Understanding the cultural nuances and contextual variations inherent in Hinglish expressions is essential for accurate sentiment analysis. Future research will focus on incorporating cultural embeddings and context-aware features into the transformer-based models to capture the subtleties of Hinglish sentiment expressions effectively.

Real-world Applications and Deployment: The successful development and validation of Hinglish sentiment analysis models can pave the way for their deployment in real-world applications, such as customer feedback analysis, social media monitoring, and e-commerce recommendation systems tailored to multilingual audiences. This expansion into multilingual sentiment analysis markets opens up new avenues for business intelligence and consumer engagement across diverse linguistic communities.

Continuous Innovation and Adaptation: As language evolves and new linguistic phenomena emerge, ongoing research and innovation will be essential to adapt and refine transformer-based models continually. Collaborative efforts with linguists, data scientists, and industry stakeholders can foster a dynamic research ecosystem dedicated to advancing multilingual sentiment analysis technologies.

In summary, the future scope of enhancing code-mixed sentiment analysis through a transformer-based approach extends beyond English to encompass Hinglish and potentially other multilingual contexts. This expansion promises to enrich our understanding of linguistic diversity and sentiment expression across different cultures, laying the groundwork for more inclusive and effective natural language processing solutions in the global marketplace.

7. CONCLUSION

In summary, our study aims to tackle the difficulties associated with sentiment analysis for languages that have mixed codes, with a particular emphasis on

Hindi–English code-mixed texts. The extant work largely focuses on English sentiment analysis, ignoring the particular subtleties presented by multilingual code-mixed data. We provided an extensive methodology and experimental setting, and we suggested a transformer-based strategy that makes use of the XLMR cross-lingual embedding model in order to close this gap.

The first acknowledgement made in the research is the lack of annotated data available for code-mixed sentiment analysis. We prioritise dataset augmentation in order to get over this restriction, producing an emotion-annotated Hindi–English (Hinglish) code-mixed dataset. This dataset provides an essential basis for training and assessing the efficacy of our suggested methodology.

Our transformer-based model introduces a multitask framework by addressing sentiment detection and emotion recognition at the same time. Task-specific data is used to refine the XLMR model and improve its comprehension of the subtleties of code-mixed sentiment. A crucial component is transfer learning, which enables the model to take use of prior information and modify it to fit the specifics of the code-mixed language environment.

Robust hardware and software configuration are used in the experimental setting to provide effective model training and assessment. Utilising a strong GPU, large amounts of RAM, and a CPU with excellent speed allows for the best possible use of computing resources. Model creation is made easier by natural language processing tools and deep learning frameworks like TensorFlow or PyTorch. Our method relies heavily on the XLMR cross-lingual embedding model, which facilitates transfer learning for improved performance.

Our transformer-based model adopts a multitask framework that simultaneously tackles sentiment detection and emotion recognition. By leveraging task-specific

data, we fine-tune the XLMR model to enhance its understanding of the nuances in code-mixed sentiment. A pivotal aspect of our approach is transfer learning, allowing the model to leverage prior knowledge and adapt it to the intricacies of the code-mixed language context.

Our model is rigorously tested against baseline models during the experimental process, demonstrating its superiority in code-mixed sentiment analysis. F1-Score, accuracy, and precision-recall curves are examples of performance indicators that offer quantitative insights into the efficacy of the model. In order to guarantee the model's adaptability to various code-mixed datasets and language settings, we also place a strong emphasis on scalability testing.

We show that our suggested method is practically applicable by assessing its performance independently of ensemble methods. This demonstrates that the model is ready for practical application in natural language processing tasks, maintaining an efficient and ready-for-production NLP model.

Our model undergoes rigorous testing against baseline models during the experimental phase, showcasing its superiority in code-mixed sentiment analysis. Performance indicators such as F1-Score, accuracy, and precision-recall curves provide quantitative insights into the model's effectiveness. Additionally, we prioritize scalability testing to ensure the model's adaptability across diverse code-mixed datasets and language contexts.

Through addressing data scarcity, leveraging multitask learning, incorporating transfer learning, and conducting comprehensive experiments, we have demonstrated the effectiveness and practicality of our transformer-based approach. This not only enhances our understanding of sentiment analysis in multilingual contexts but also lays the groundwork for more accurate and reliable

sentiment analysis solutions in code-mixed languages like Hindi–English. As the field continues to evolve, our work represents a foundational step toward bridging the gap between monolingual sentiment analysis and the complexities of multilingual, code-mixed data, opening new avenues for research and innovation in this domain.

In conclusion, our work advances the field by providing a revolutionary approach to the problems associated with code-mixed sentiment analysis. In addition to improving our knowledge of sentiment in multilingual circumstances, the transformer-based approach that has been suggested and is backed by a strong methodology and experimental setup also establishes a standard for future work on improving sentiment analysis for code-mixed languages.

By addressing the scarcity of annotated data, leveraging multitask learning, incorporating transfer learning, and conducting comprehensive experimentation, we have demonstrated the effectiveness and practical applicability of our transformer-based approach. This not only enhances our understanding of sentiment analysis in multilingual contexts but also paves the way for more accurate and reliable sentiment analysis solutions for code-mixed languages like Hindi–English. As the field continues to evolve, our work serves as a foundational step towards bridging the gap between monolingual sentiment analysis and the complexities of multilingual, code-mixed data, opening avenues for further research and innovation in this domain.

8. References

- [1] C. Hoffman, An Introduction to Bilingualism 4th impression, Longman Ltd, UK, 1996.
- [2] P. Ekman, E.R. Sorenson, W.V. Friesen, Pan cultural elements in facial displays

of emotion, *Science* 164 (3875) (1969) 86–88.

[3] O. Bălan, G. Moise, L. Petrescu, A. Moldoveanu, M. Leordeanu, F. Moldoveanu, Emotion classification based on biophysical signals and machine learning techniques, *Symmetry* 12 (1) (2020) 21.

[4] S.-Y. Chen, C.-C. Hsu, C.-C. Kuo, L.-W. Ku, et al., Emotionlines: An emotion corpus of multi-party conversations, 2018, arXiv preprint arXiv:1802.08379.

[5] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020*, pp. 8440–8451, <http://dx.doi.org/10.18653/v1/2020.acl-main.747>.

[27] H. Wang, D. P. Tobon V., M. S. Hossain, and A. El Saddik, “Deep learning (DL) enabled system for emotional big data,” *IEEE Access*, vol. 9, pp. 116073–116082, 2021.

[28] Y. Miao, H. Dong, J. M. A. Jaam, and A. E. Saddik, “A deep learning system for recognizing facial expression in real-time,” *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 2, pp. 1–20, May 2019.

[29] V. Athanasiou and M. Maragoudakis, “A novel, gradient boosting framework for sentiment analysis in languages where NLP resources are not plentiful: A case study for modern Greek,” *Algorithms*, vol. 10, no. 1, p. 34, 2017.

[30] R. Abaalkhail, F. Alzamzami, S. Aloufi, R. Alharthi, and A. El Saddik, “Affectional ontology and Conf. Smart Multimedia. Cham, Switzerland: Springer, 2018, pp. 15–28.

[6] A. Agarwal, P. Bhattacharyya, Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be

classified, in: Proceedings of the International Conference on Natural Language Processing, Vol. 22, ICON, 2005.

[7] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[8] B.G. Patra, D. Das, A. Das, Sentiment analysis of code-mixed Indian languages: An overview of SAIL_Code Mixed Shared Task@ ICON-2017, 2018, arXiv preprint arXiv:1803.06745.

[9] S. Ghosh, S. Ghosh, D. Das, Sentiment identification in code-mixed social media text, 2017, CoRR abs/1707.01184, URL <http://arxiv.org/abs/1707.01184>.

[10] A. Bohra, D. Vijay, V. Singh, S.S. Akhtar, M. Shrivastava, A dataset of Hindi-English code-mixed social media text for hate speech detection, in: Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in social media, 2018, pp. 36–41.

[11] A. Joshi, A. Prabhu, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of Hindi-English code-mixed text, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2482–2491.

[12] K. Hashimoto, C. Xiong, Y. Tsuruoka, R. Socher, A joint many-task model: Growing a neural network for multiple NLP tasks, 2016, arXiv preprint arXiv:1611.01587.

[13] H. Wang, D. Can, A. Kazemzadeh, F. Bar, S. Narayanan, A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle, in: Proceedings of the ACL 2012 System Demonstrations, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 115–120, <https://www.aclweb.org/anthology/P12-3020>.

- [14] S. Shekhar, D. Sharma, M. Beg, an effective biLSTM word embedding system for analysis and identification of language in code-mixed social media text in English and Roman Hindi, *Compute. Sist.* 24 (2020) <http://dx.doi.org/10.13053/cys-24-4-3151>.
- [15] P. Mathur, R. Sawhney, M. Ayyar, R. Shah, did you offend me? Classification of offensive tweets in Hinglish language, in: *Proceedings of the 2nd Workshop on Abusive Language Online, ALW2*, 2018, pp. 138–148.
- [16] R. Caruana, multitask learning, *Mach. Learn.* 28 (1) (1997) 41–75.
- [17] A. Kumar, A. Ekbal, D. Kawahara, S. Kurohashi, Emotion helps sentiment: A multi-task model for sentiment and emotion analysis, in: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, IEEE, 2019, pp. <http://dx.doi.org/10.1109/IJCNN.2019.8852352>.
- [18] M.S. Akhtar, D.S. Chauhan, D. Ghosal, S. Poria, A. Ekbal, P. Bhattacharyya, Multi-task learning for multi-modal emotion recognition and sentiment analysis, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 370–379, <http://dx.doi.org/10.18653/v1/n19-1034>.
- [19] M.S. Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, S. Kurohashi, All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework, *IEEE Trans. Affect. compute.* 13 (1) (2022) 285-297, <http://dx.doi.org/10.1109/TAFFC.2019.2926724>.
- [20] S. Ghosh, A. Ekbal, P. Bhattacharyya, A multitask framework to detect depression, sentiment and multi-label emotion from suicide notes, *Cogn. compute.* (2021) 1–20.
- [21] S.R. Sane, S. Tripathi, K.R. Sane, R. Mamidi, Stance detection in code-mixed

Hindi-English social media data using multi-task learning, in: Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2019, pp. 1–5.

[22] F. Alzamzami, M. Hoda, and A. El Saddik, “Light multimedia dataset for sentiment analysis,” in Proc. Int. gradient boosting machine for general sentiment classification on short texts: A comparative evaluation,” IEEE Access, vol. 8, pp. 101840–101858, 2020.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16×16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[24] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim, “AdaViT: Adaptive vision transformers for efficient image recognition,” 2021, arXiv:2111.15668.

[25] F. Alzamzami and A. El Saddik, “Monitoring cyber SentiHate social behavior during COVID-19 pandemic in North America,” IEEE Access, vol. 9, pp. 91184–91208, 2021.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[31] Harrison Rainie, Janna Quitney Anderson, and Jonathan Albright. 2017. The future of free speech, trolls, anonymity and fake news online. Pew Research Center Washington, DC.

[32] Tharindu Ranasinghe and Marcos Zampieri. 2021. An evaluation of multilingual offensive language identification methods for the languages of india. Information 12, 8 (2021), 306.

[33] Diana Rieger, Anna Sophie Kümpel, Maximilian Wich, Toni Kiening, and

Georg Groh. 2021. Assessing the Extent and Types of Hate Speech in Fringe Communities: A Case Study of Alt-Right Communities on 8chan, 4chan, and Reddit. *Social Media+ Society* 7, 4 (2021), 20563051211052906.

[34] Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian twitter corpus of hate speech against immigrants. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[35] Arushi Sharma, Anubha Kabra, and Minni Jain. 2022. Ceasing hate with MoH: Hate Speech Detection in HindiEnglish code-switched language. *Information Processing & Management* 59, 1 (2022), 102760.

[36] Michael Simonson. 2017. Social Media and Online Learning: Pros and Cons. *Distance Learning* 14, 4 (2017), 72ś71.

[37] Elise Fehn Unsvåg and Björn Gambäck. 2018. The effects of user features on twitter hate speech detection. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*. 75ś85.

[38] William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*. 19ś26.