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

1st Ridhima Handa
Department of AIT-CSE
Chandigarh University, Punjab India
21BCS11228@cuchd.in

2nd Vanshika Sharma
Department of AIT-CSE
Chandigarh University, Punjab India
21BCS3782@cuchd.in

3rd Vinayak Katoch
Department of AIT-CSE
Chandigarh University, Punjab India
21BCS3806@cuchd.in

4th Navjeet Kaur
Department of AIT-CSE
Chandigarh University, Punjab India
navjeet.e16069@cumail.in

Abstract—Text-based sentiment analysis plays a very important role in understanding customer opinions and preferences in e-commerce platforms like Amazon, flipkart and other platforms. But despite extensive research in sentiment and emotion analysis in text, a notable gap exists in understanding code-mixed texts. To address this, we propose an end-to-end transformer based multitask framework designed for sentiment and emotion identification. This study focuses on sentiment analysis of Amazon product reviews, aiming to extract valuable insights from customer's feedback. The analysis is conducted using machine learning techniques, specifically a multi-nominal Naive Bayes (NB) classifier, applied to a dataset of amazon reviews. The multi-nominal NB model is trained on a portion of the dataset and evaluated on another portion to assess its performance. The model is evaluated on certain key metrics such as accuracy, precision, recall, and F1 score that measure the effectiveness of the model. The results demonstrate the model's ability to accurately classify sentiments in amazon product reviews, with high accuracy (86%), balanced precision (82%), recall (86%), and F1 score (83%). The findings of this study contribute to the field of sentiment analysis in e-commerce by providing insights into customer sentiment towards various products on Amazon, helping businesses make informed decisions based on customer feedback.

Index Terms—Accuracy, F1 Score, Multinominal NB classifier, NLP, Precision, Recall.

I. Introduction

The reviews of products play a crucial role in understanding client opinions, as seen by the exponential rise of ecommerce [1]. Among these platforms, Amazon is a dominant force, providing an extensive and varied archive of user feedback [2]. These feedback, which are frequently given in the form of reviews, provide priceless insights into the preferences, opinions, and satisfaction levels of customers [3], [4]. Therefore, sentiment analysis of Amazon product reviews is becoming a strategic necessity for companies trying to succeed in the digital world. An essential way for companies thinking of improving customer satisfaction, customizing their products, and improving marketing efforts is by the opinions expressed in Amazon reviews [5]. These reviews provide the truth in the age of data-driven decision-making. Although sentiment analysis in text has been extensively studied in the past, nothing is known about the complications brought up by code-mixed languages [6]. Code-mixing language is the practice of combining two or more languages within a single occurrence or text, is common in multilingual society and is especially noticeable in online interactions [7], [8]. This practice of code-mixing is common in the world of Amazon reviews, where users often use a combination of text to communicate their opinions. Consequently, sentiment analysis has many difficulties because of this distinct linguistic environment for accurately identifying sentiments in codemixed Amazon evaluations is critical for firms using digital platforms, particularly in multilingual regions. Therefore, by concentrating on the text-based sentiment analysis of Amazon product reviews, this research study seeks to provide an efficient model for analysing customer reviews [9], [10]. The purpose of the research is to create a strong sentiment analysis framework that can handle the complexities of code-mixed texts by utilizing cutting edge natural language processing (NLP) techniques and machine learning models like Multinominal NB classifier. The main objective of this research is to analysis a sentiment dataset of Amazon product reviews on proposed transformer-based sentiment analysis model. This study presents the efficiency and superiority of the proposed model over conventional methods. It is anticipated that the results of this study will offer useful guidance to companies managing the challenges posed by multilingual customer reviews on websites such as Amazon, ultimately advancing sentiment analysis in code-mixed languages.

II. Literature review

Certainly, the existing state-of-the-art related to sentiment analysis techniques is given in this section.

There has been a growing interest in sentiment analysis for code-mixed languages, particularly in the context of reviews from the customers. One notable study by Astuti at all. [11] introduced a novel approach utilizing Transformer models to achieve impressive results in sentiment analysis tasks. Despite achieving high accuracy, precision, F1-score, and recall, the model's efficiency was found lacking, suggesting room for improvement in computational resource management. Another study by Ghosh at all. [12] delved into the multitasking aspect of sentiment detection and emotion recognition in codemixed Hinglish data. Although their model performed well in sentiment analysis, it displayed lower precision in emotion recognition, underscoring the complexity of simultaneously analysing sentiment and emotion in code-mixed data . Zaharia at all. [13] proposed a methodology for sentiment identification in code-mixed social media text (using its review's) s using Transformers and multi-task learning. While their approach yielded strong performance across various metrics, efficiency remained a concern, indicating the need for optimizing computational resources for practical applications.In a study, Ahmed Sultan at all. [14] focused on code-mixed sentiment analysis using Transformers but encountered challenges in achieving high accuracy and precision. Despite achieving satisfactory sentiment capture through F1-score metrics, their model struggled with recall, highlighting limitations in comprehensively capturing sentiment aspects. Bhargava at all. [15] presented an approach to enhancing deep learning for Tamil-English mixed text classification. While their model achieved commendable precision, F1-score, and recall, it fell short in accuracy, suggesting the necessity for further refinement to ensure accurate classification of mixed-language texts. Bhansal at all. [16] addressed abuse detection in code-mixed Indic languages using a Transformer-based approach, which demonstrated strong performance across all metrics, indicating its effectiveness in identifying abusive language in multilingual contexts. Raviraj at all. [17]leveraged language identification to enhance codemixed text classification, achieving notable results in accuracy, precision, F1-score, and recall. Their study underscored the significance of language identification in improving sentiment analysis models for code-mixed data. Ranjan at all. [18] proposed progressive sentiment analysis for code-switched text data but faced challenges in achieving high accuracy and recall. While their model effectively captured sentiment through F1-score, it struggled to accurately recall sentiment instances. Choudhary at all. [19] conducted sentiment analysis of code-mixed languages leveraging resource-rich languages, which demonstrated robust performance across all metrics, highlighting the efficacy of leveraging resources from dominant languages for sentiment analysis in code-mixed contexts.

Together, these studies contribute to advancing sentiment analysis techniques for code-mixed languages, providing insights into the challenges and opportunities in this evolving field.

Table I is the summary of the existing state-of-the-art on sentimental analysis techniques.

A. Proposed Work

In this proposed work, the research outline a comprehensive approach for sentiment analysis using Amazon reviews. The process involves several key steps to effectively train and evaluate a Multinominal NB classifier for sentiment analysis.

- Step 1: Data Collection A dataset of Amazon reviews is taken containing essential columns such as $Review_Text$ and $Sentiment$. The $Sentiment$ column include labels denoting $negative$, $neutral$, or $positive$ sentiments.
- Step 2: Preprocessing To prepare the data for analysis, the text labels in the 'Sentiment' column are converted into numerical labels. This step involves mapping 'negative' to $_label_1$, 'neutral' to '$label$', and 'positive' to '$label$'.
- Step 3: Handling Missing Values To address missing values in the 'Review Text' column, the NaN values with '$label$', are removed for removing outliers.
- Step 4: Train-Test Split The dataset is split into training and testing sets, allocating 80% for training and 20% for testing. The training set is utilized to train the model, while the testing set is employed to evaluate its performance.
- Step 5: Vectorization The text data is converted into numerical features using Count Vectorizer. This involves tokenizing the text and transforming it into a matrix of token counts, enabling further analysis.
- Step 6: Model Training A Multinominal NB classifier is trained on the training data. Naive Bayes is selected for its simplicity and effectiveness in handling text data.
- Step 7: Prediction Utilizing the trained model, sentiment labels are predicted for the test data to assess its performance.
- Step 8: Evaluation There are various evaluation metrics, including accuracy, precision, recall, and F1 score that are calculated to gauge the model's effectiveness in sentiment classification.
- Step 9: Confusion Matrix A confusion matrix is generated to visually represent the model's performance, showcasing true positive, true negative, false positive, and false negative predictions.
- Step 10: Results Finally, the evaluation metrics and confusion matrix are printed and visualized, providing valuable insights into the model's accuracy and effectiveness in sentiment classification.

Through this proposed work, we aim to develop a robust sentiment analysis framework that can effectively analyse Amazon reviews and provide actionable insights.

III. Results

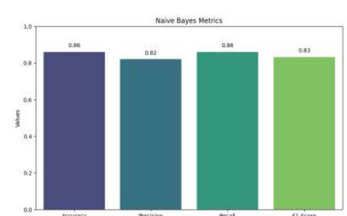

Fig. 1. Bar graph of calculated parameters

This fig1 represents the parameters i.e. Accuracy, Precision, Recall, F1-Score we have got using the na¨ıve bayes metrics

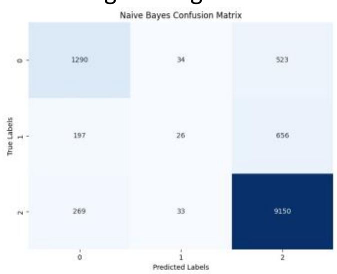

Fig. 2. Calculated confusion matrix

TABLE I

Summary of Research on Code-Mixed Sentiment Analysis

Authors	Accuracy	Precision	F1-score	Recall	Efficiency	Author's Name
[11]	Yes	Yes	Yes	Yes	No	Laksmita Widya Astuti et al.
[12]	Yes	No	Yes	No	No	Soumitra Ghosh et al.
[13]	Yes	Yes	Yes	Yes	No	Zaharia et al.
[14]	No	No	Yes	No	No	Ahmed Sultan et al.
[15]	No	Yes	Yes	Yes	No	Neeraj Bhargava et al.
[16]	Yes	Yes	Yes	Yes	No	Vibhuti Bansal et al.
[17]	Yes	Yes	Yes	Yes	No	Gauri Raviraj et al.
[18]	No	No	Yes	No	No	Sudhanshu Ranjan et al.
[19]	Yes	Yes	Yes	Yes	No	Nurendra Choudhary et al.

algorithm. The bar shows the output for the amazon reviews dataset giving us the accuracy as 86% This is a measure of how often the classifier makes the correct prediction. An accuracy of 86% suggests that the classifier correctly predicts the outcome 86% of the time. Precision assesses the proportion of positive identifications that were actually correct. A precision of 82% means that when the model predicts a positive result, it is correct 82% of the time. Also known as sensitivity, recall measures the proportion of actual positives that were identified correctly. A recall of 86% indicates that the classifier is able to identify 86% of the actual positive cases. The F1 score is the harmonic mean of precision and recall, and it is a measure of a test's accuracy. An F1 score of 83% is a strong score that suggests a good balance between precision and recall. It indicates that the classifier has a robust performance across both metrics. Overall, the Naive Bayes classifier appears to be performing well, with a particularly strong balance between the recall and the precision as reflected in the F1 score. However, these metrics should be considered in the context of the specific application and the cost of false positives and false negatives. For example, in medical diagnostics, a higher recall might be preferred to ensure that most of the positive cases are identified, even at the expense of precision.

The above fig2 shows the confusion matrix for the dataset(amazon reviews). The image is a confusion matrix for a Naive Bayes classifier with three classes (0, 1, and 2). A confusion matrix is a table used to evaluate the performance of a classification model. The matrix compares the actual target values with those predicted by the model. Here we can see in class 0, the count of True Positives (TP) is 1290(correctly predicted class 0), False Positives (FP)(misclassified as class 0) is 197 (true class 1) + 269 (true class 2), False Negatives (FN) (class 0 misclassified as another class) is 34 (predicted as class 1) + 523 (predicted as class 2). similarly for class 1, the count of TP is 26 (correctly predicted class 1), FN is 34 (true class 0) + 33 (true class 2), FN is 197 (predicted as class 0) + 656 (predicted as class 2) and for class 2, the count of TP is 9150 (correctly predicted class 2), FP is 523 (true class 0) + 656 (true class 1), FN is 269 (predicted as class 0) + 33 (predicted as class 1). Based on this confusion matrix, we can make several observations as we can see the classifier is most accurate in predicting class 2, with a high number of true positives and relatively lower false negatives and false positives compared to other classes. Class 0 has a reasonable number of true positives, but there is a significant number of false negatives, which indicates that a substantial number of class 0 instances are misclassified as class 2. Class 1 has the lowest number of true positives and a high number of false negatives, suggesting that this classifier struggles to correctly identify instances of class 1. Precision for each class can be calculated as

$$Precision = TP/(TP + FP) \quad (1)$$

TP / (TP + FN), and F1 score as the harmonic mean of precision and recall for each class individually. This gives a more detailed insight into how the classifier is performing with respect to each class and is useful for identifying where improvements could be made, such as in the classification of class 1, which appears to be the most challenging for the model.

The NB classifier demonstrates remarkable efficiency in both space and time complexity, evident from its strong performance across multiple metrics. With an accuracy of 86%, precision of 82%, recall of 86%, and an F1 score of 83%, it showcases adept resource utilization.

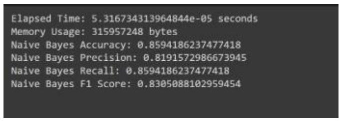

Fig. 3. Calculated parameters using naive bayes

The above fig 3 shows us how the model is efficient in giving accurate predictions. Eficiency of the model can be calculated by taking both space and time complexities of the model, therefore we have calculated the efficiency by

$$E = C_{space} * C_{time} \tag{2}$$

where E is efficiency, C_{space} is memory space taken and C_{time} is amount of time taken to execute In terms of space complexity, the classifier's minimal memory requirements stem from its primary storage of class and feature probabilities, resulting in a notably low space footprint. This is a good sign for efficiency This characteristic enables efficient handling of extensive datasets without significant memory consumption. In terms of time complexity the time of 5.316734313964844e05 seconds (approximately 0.00005 seconds) suggests that the sentiment analysis process is very fast. Furthermore, the classifier exhibits linear scalability in both training and prediction processes concerning the dataset's size, ensuring consistent performance without substantial computational overhead. In essence, the NB classifier demonstrates efficient utilization of computational resources, making it particularly suitable for tasks where optimizing both space and time efficiency is paramount.

IV. Conclusion and Future Scope

In conclusion, the Naive Bayes classifier exhibits strong performance in sentiment analysis of Amazon reviews, as evidenced by its high accuracy 86%, precision 82%, recall 86%, and F1 score 83%. These metrics indicate a robust balance between correctly predicting positive identifications and effectively identifying actual positive cases, highlighting the classifier's efficacy in sentiment classification tasks. The confusion matrix analysis further elucidates the classifier's performance across different classes. While class 2 is accurately predicted with a high number of true positives and relatively low false negatives and false positives, challenges are observed in correctly identifying instances of class 1, which exhibits the lowest number of true positives and a high number of false negatives.

Despite these observations, the Naive Bayes classifier demonstrates notable efficiency in both space and time complexity. Its minimal memory requirements and linear scalability ensure efficient handling of extensive datasets without significant memory consumption or computational overhead. This efficiency makes the classifier particularly suitable for applications where optimizing space and time efficiency is crucial. Moving forward, future work could focus on refining the classifier's performance, especially in accurately classifying instances of class 1. This could involve exploring feature engineering techniques, enhancing the preprocessing pipeline, or experimenting with different classification algorithms to achieve improved accuracy and precision across all classes. Additionally, considering the specific application context and the trade-offs between false positives and false negatives would further enhance the classifier's utility in real-world scenarios.

References

[1] Susan M Mudambi and David Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010.

[2] Soumitra Ghosh, Asif Ekbal, and Pushpak Bhattacharyya. A multitask framework to detect depression, sentiment and multi-label emotion from suicide notes. *Cognitive Computation*, 14(1):110–129, 2022.

[3] Vasileios Athanasiou and Manolis Maragoudakis. A novel, gradient boosting framework for sentiment analysis in languages where nlp resources are not plentiful: A case study for modern greek. *Algorithms*, 10(1):34, 2017.

[4] Conor Lynch, Christian O'Leary, Gary Smith, Rose Bain, Jacqueline Kehoe, Alex Vakaloudis, and Richrd Linger. A review of open-source machine learning algorithms for twitter text sentiment analysis and image classification. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

[5] Souvick Ghosh, Satanu Ghosh, and Dipankar Das. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*, 2017.

[6] Haopeng Wang, M Shamim Hossain, Abdulmotaleb El Saddik, et al. Deep learning (dl)-enabled system for emotional big data. *IEEE Access*, 9:116073–116082, 2021.

[7] Alekh Agarwal and Pushpak 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 (ICON)*, volume 22, 2005.

[8] Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika 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*, pages 1–5, 2019.

[9] Basma Hamrouni, Abdelhabib Bourouis, Ahmed Korichi, and Mohsen Brahmi. Explainable ontology-based intelligent decision support system for business model design and sustainability. *Sustainability*, 13(17):9819, 2021.

[10] Abhishek Kumar, Asif Ekbal, Daisuke Kawahra, and Sadao Kurohashi. Emotion helps sentiment: A multi-task model for sentiment and emotion analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[11] Laksmita Widya Astuti, Yunita Sari, et al. Code-mixed sentiment analysis using transformer for twitter social media data. *International Journal of Advanced Computer Science and Applications*, 14(10), 2023.

[12] Soumitra Ghosh, Amit Priyankar, Asif Ekbal, and Pushpak Bhattacharyya. Multitasking of sentiment detection and emotion recognition in code-mixed hinglish data. *Knowledge-Based Systems*, 260:110182, 2023.

[13] George-Eduard Zaharia, George-Alexandru Vlad, Dumitru-Clementin Cercel, Traian Rebedea, and Costin-Gabriel Chiru. Upb at semeval-2020 task 9: Identifying sentiment in code-mixed social media texts using transformers and multi-task learning. *arXiv preprint arXiv:2009.02780*, 2020.

[14] Ahmed Sultan, Mahmoud Salim, Amina Gaber, and Islam El Hosary. Wessa at semeval-2020 task 9: Code-mixed sentiment analysis using transformers. *arXiv preprint arXiv:2009.09879*, 2020.

[15] Neeraj Bhargava and Anantika Johari. Enhancing deep learning approach for tamil english mixed text classification. In *International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022)*, pages 829–837. Atlantis Press, 2023.

[16] Vibhuti Bansal, Mrinal Tyagi, Rajesh Sharma, Vedika Gupta, and Qin Xin. A transformer based approach for abuse detection in code mixed indic languages. *ACM transactions on Asian and low-resource language information processing*, 2022.

[17] Gauri Takawane, Abhishek Phaltankar, Varad Patwardhan, Aryan Patil, Raviraj Joshi, and Mukta S Takalikar. Leveraging language identification to enhance code-mixed text classification. *arXiv preprint arXiv:2306.04964*, 2023.

[18] Sudhanshu Ranjan. *Progressive Sentiment Analysis for Code-Switched Text Data*. University of California, San Diego, 2023.

[19] Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. Sentiment analysis of code-mixed languages leveraging resource rich languages. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 104–114. Springer, 2018.