Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 24, Issues 4, pp. 873–882, DOI 10.12694/scpe.v24i4.2375

MULTILINGUAL CODE-MIXED SENTIMENT ANALYSIS IN HATE SPEECH

TULIKA RANJAN[*], ANISH SINGH[†], RINA KUMARI[‡], SUJATA SWAIN[§], ANJAN BANDYOPADHYAY[¶], AND AJAYA KUMAR PARIDA[‖]

Abstract. Sentiment analysis discovers the emotion expressed in a text. It helps in analyzing the product reviews, customer feedback and survey responses. Researchers have developed various algorithms for this purpose, however, they have majorly focused only on the sentiment analysis in English language. Although, few works are available for Hindi and multilingual sentiment analysis, however, these works are not efficient enough to perform sentiment analysis in code-mixed languages. To overcome the limitation of the existing works, this paper presents a multilingual code-mixed language model which identifies the sentiments of the hate speech dataset extracted from Twitter. As the hate speech dataset with sentiment labels are not available, we first collect the data from Twitter. After that we label the data using a transformer-based pretrained sentiment analysis model trained on a large corpus of tweets in multiple languages. We pass our collected data as test data to this model and predict the sentiment labels. Now, we train six different machine learning models to perform our own task i.e sentiment analysis for multilingual code-mixed hate speech dataset. The machine learning models perform well across multiple languages and also code-mixed languages. In future, it can be easily adapted to different classification tasks based on code-mixed languages. The results yield that hate speech invokes negative sentiment whereas non-hate speech reflects either positive or neutral sentiment.

Key words: Code-Mixed, Multilingual text data, Sentiment analysis, Hate speech, Natural language processing, Machine learning

1. Introduction. The internet has facilitated communication and sharing of opinions but also enabled the spread of hate news that target and harms individuals and communities based on their appearance, religion, or sexual orientation. India, with its diverse linguistic and cultural landscape, has become a hotbed for spreading hatred through online platforms. Prior investigations on hate speech detection have majorly focused on high-resource languages like English, but the prevalence of code-mixing in Indian languages like Hindi-English (Hinglish) calls for more attention to detect hate speech in such multilingual contexts [5, 6]. Code-mixing [13, 10] refers to the practice of using words and phrases from multiple languages in a single sentence or expression. The dissemination of hate news in a multi-lingual society like India is a challenging issue due to a lack of media regulation and verification. Hate news can manipulate users for financial, religious, or political purposes and harm society as a whole. Recent incidents in India, including hate-mongering during political rallies and racial discrimination during the COVID-19 pandemic, have underscored the need to prevent the transmission of hate speech through online platforms. Detecting hate speech in code-mixed text requires the development of AI models that can accurately interpret and identify hateful content [8]. The ability to detect and prevent hate speech in multilingual contexts has wider implications for promoting diversity, equity, and inclusion in online spaces. The development of such models can also aid in the creation of safer and more welcoming online communities for people from diverse linguistic and cultural backgrounds. Table 1.1 depicts some samples from the collected dataset.

Below examples list some instances of Hindi-English code-mixed text. It also discusses the translated version of the instance in English.

[*]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (tulikaranjan2002@gmail.com).
[†]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (anishsingh1452002@gmail.com).
[‡]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (rina.kumarifcs@kiit.ac.in).
[§]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (sujata.swainfcs@kiit.ac.in).
[¶]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (anjan.bandyopadhyay@kiit.ac.in).
[‖]School of Computer Engineering, Kalinga Institute of Industrial Technology, India (ajaya.paridafcs@kiit.ac.in).

Table 1.1: Samples of dataset with sentiment labels. Here Neg, Pos and Neut represent Negative, Positive and Neutral, respectively

Sentence	**Label0 (Neg)**	**Label1 (Neut)**	**Label2 (Pos)**
Mera Kaam Office mai hai		✓	
Aaj me woh moive dekhne jaa rahi hu.		✓	
We hate him utna hi jitna we hate you.	✓		
Mujhe maths se nafrat hai. I wish it didn't exist	✓		
He loves to play Gilli danda			✓

Examples of Code Mixed
Text-instance-1 : *"Mera kaam office mein hai."*
Translation : *"My work is in the office."*
Hate Speech Label : *"Non Hate"*
Sentiment Label : *"Neutral"*
Text-instance-2 : *"Maine apna homework kiya hai."*
Translation : *"I have done my homework."*
Hate Speech Label : *"Non Hate"*
Sentiment Label : *"Neutral"*
Text-instance-3 : *"Tumne mujhe email kiya tha kya?"*
Translation *"Did you send me an email?"*
Hate Speech Label : *"Non Hate"*
Sentiment Label : *"Neutral"*

Examples of Code Mixed Hate Speech
Text-instance-1 : *"We hate him utna hi jitna we hate you."*
Translation : *"We hate him as much as we hate you."*
Hate Speech Label : *"Hate"*
Sentiment Label : *"Negative"*
Text-instance-2 : *"Mujhe maths se nafrat hai. I wish it didn't exist."*
Translation : *"I hate maths. I wish it did not exist."*
Hate Speech Label : *"Hate"*
Sentiment Label : *"Negative"*
Text-instance-3 : *"He loves to play Gili danda."*
Translation : *"He loves to play Gili danda."*
Hate Speech Label : *"Non Hate"*
Sentiment Label : *"Positive"*

In light of the above examples, we can analyze that hate speech provoke negative sentiment whereas non hate provoke either positive or neutral sentiments. The negative sentiments provokes people to spread the hate speech on social media platforms which negatively impacts the individual and society and led to major harm. Therefore, It is important to prevent the spread of hate speech on social media and for this purpose sentiment analysis can be proved a key aspect. By motivating this idea, we train various machine learning models which perform sentiment analysis in the multilingual code-mixed hate speech collected from twitter. As per our investigations, this is a novel approach for multilingual code-mixed languages more specifically in hate speech.

1.1. Problem Statement And Specification Requirement. The problem statement of the Multilingual Sentiment analysis model is to perform sentiment analysis on code-mixed multilingual hate speech tweets.

This requires the model to be able to accurately classify tweets as positive, negative, or neutral, regardless of the language used in the tweet.

To accomplish this, the model needs to meet the following requirements:

1. Process text in multiple languages
2. Accurately identify sentiment polarity in tweets
3. Perform well on a variety of datasets and languages
4. Handle noisy data and non-standard language usage in tweets
5. Efficient and salable enough to be used in real-world applications

We fulfill the above mentioned requirements in the following subphases:

1. Problem definition: We first define the problem statement and goals of the investigation which includes gathering of the specific objectives of this investigation and the target audience for the sentiment analysis. Here, we provide hate and non hate speech data in text format to various machine learning models as input and the model detects the sentiment (positive, negative or neutral) of those sentences.
2. Data source identification: In this subphase, we identify the data sources which are used to collect the data for the sentiment analysis. For our model, we extract the data from twitter.
3. Data cleaning and preparation: Data cleaning is essential for accurate sentiment analysis. In this stage, we remove the irrelevant data, perform text normalization, and prepare the data for analysis.
4. Model selection: Since our collected dataset is not labeled with the sentiment classes, after data pre-processing, we select a pre-trained sentiment analysis model to get the sentiment labels. After that, we select six machine learning models to perform sentiment analysis in code-mixed hate speech dataset.
5. Model training: After selecting the appropriate pre-trained model, we pass our own collected data as the test data to the model and the model predicts the sentiment labels for the data instance. Further, we train six machine learning models using the labeled code-mixed multilingual hate speech dataset (our dataset). We perform cross-validation to evaluate the model's performance and make any necessary adjustments. Further, we consider precision, recall, F1-score, accuracy, model complexity, and computational resources as the measuring factors to measure the efficiency of the models.
6. Model testing: After training, we test the models on a separate data set to evaluate their performance.

2. Literature Review. People share their thoughts in social media. However, they are not only using English language but also they mixed their own mother-tongues language with it. As India has 22 different languages. So, social media contents moreover are the code mixed information. There are different studies that focus on the developing a model for the code mixed sentiment analysis in recognizing hate speech. The study presented in [4] demonstrates the hate speech detection problem in code-mixed texts. In this work, the authors have first developed a Hindi-English code-mixed dataset which contains tweets posted online on Twitter. The tweets are annotated with the language at word level and the class they belong to (Hate Speech or Normal Speech). The authors have also proposed a supervised classification framework for hate speech detection in the text using different lexicon-based, character level and word level features. This task holds significant relevance for numerous applications, such as cyberbullying investigation, sentiment analysis, and examining socio-political controversies. The work proposed in [2], discusses the hate Speech detection task for Code-mixed text in Tamil and English languages. They first developed a dataset with 10,000 Tamil-English code-mixed texts collected from Twitter. Also, each text is annotated with hate or non-hate text. After preparing the dataset, they have developed a synonym-based Bi-LSTM model for classifying hate and non-hate text in tweets. In [5], Hinglish hate speech detection is discussed. In this study, the text contains both Hindi and English. The proposed model[5] is based on ensemble method. It has classified the text 3 categories: Abusive, Non-Offensive and ate-Inducing. In [11], the authors have addressed the growing concern of hate speech in user-generated content, specifically on social media platforms. The authors highlight the need for automated detection of hateful content to counteract these harmful activities. They have identified hate speech within code-mixed social media text. In [12], the proposed model is detecting of hate speech text in Hindi English code mixed data. In [14, 1, 15], the authors provides a comprehensive review of studies on deep learning approaches for multilingual sentiment analysis of social media. In [8], the authors have designed a user interface based on a web browser plugin over Facebook and Twitter to visualize the aggressive comments posted on the Social media user's timelines. In [3, 9], an ensemble model for code-mixed data of hate speech classification task on Hindi-English data is

Table 3.1: Dataset distribution across various classes

Labels	Positive	Negative	Neutral
Classified in %	30.78%	38.34%	30.87%

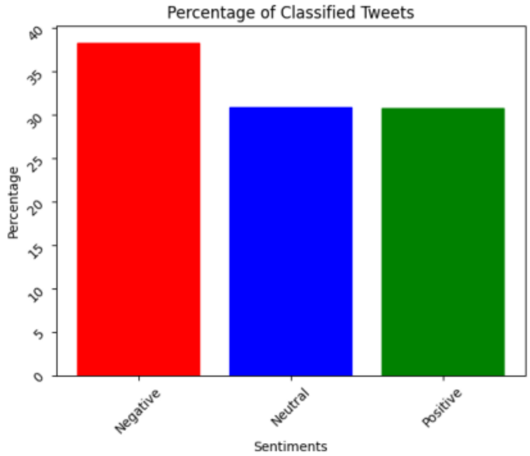

Fig. 3.1: Percentage of Classified Tweets

proposed. Also, it compares with the pretrained models and validates the proposed approach. In [7], a code-mixed English–Hindi dataset with a well-defined context is developed. It proposes context representation for conversational dialogue.

3. Overview of Dataset. The dataset utilized in our study consists of 9700 texts categorized into three classes: Positive, Negative, and Neutral. This dataset provides a solid foundation for training and evaluation of our sentiment analysis model, enabling a detailed exploration of sentiment patterns and trends within the data. The initial dataset use for training the pre-trained model was sourced from a publicly available site Kaggle[1] which consists of 1,600,000 tweets extracted using the twitter API. The dataset instance includes texts representing various sentiment categories (0 = negative, 2 = neutral, 4 = positive). These texts were thoughtfully chosen to encompass a broad range of sentiments, covering diverse topics and contexts. We use this dataset to train the machine learning models.

For the second segment of the dataset, we utilized the SNS scrape Twitter module to collect additional data. This module enabled us to retrieve texts from Twitter, specifically focusing on tweets related to our target domain. By leveraging the functionality provided by the SNS scrape Twitter module, we extracts a substantial number of tweets, capturing real-time user opinions and expressions of sentiment. We use this dataset to train the machine learning models for our own task i.e sentiment analysis on code-mixed hate speech dataset. Table 3.1 explains the distribution of the dataset across sentiment classes and Figure 3.1 shows its graphical representation. In the process of collecting the hate speech dataset from Twitter, we implemented several strategies to ensure its representativeness in terms of language diversity and code-mixed patterns. We utilized specific keywords and language filters to target relevant content, and we developed techniques to identify code-mixed language patterns within the collected data. These measures were taken to ensure that the dataset encompasses a wide range of multilingual code-mixed hate speech scenarios.

4. Methodology. We develop the multilingual code-mixed sentiment analysis model in several phases, including data collection, pre-processing, model training, and evaluation. We collect the dataset from Twitter/Kaggle and pre-process using various techniques such as tokenization, normalization, and cleaning. After

[1] https://www.kaggle.com/

preparing the dataset, we first train the various machine learning models using a generic sentiment analysis dataset (briefly discussed in dataset section) and then we fine-tune the trained model on the pre-processed dataset for the sentiment analysis on multilingual code-mixed hate speech dataset. Further, we evaluate the model using various metrics such as accuracy, precision, recall, F1-score, and confusion matrix. We also conduct the error analysis to identify the common types of errors made by the model and address them. Overall, this investigation involves several steps and techniques to develop and evaluate the model for sentiment analysis on in multilingual code-mixed hate speech Tweets. We briefly discuss all these steps in subsequent subsections.

4.1. Pre-processing. To develop any data dependant model, data preprocessing is a key task which boosts the performance of the model. Therefore, we first pre-process the dataset following techniques described below:

1. **Removing Noise:** Noise removal techniques were implemented to eliminate irrelevant or undesirable elements from the Twitter data. This involved filtering out special characters, URLs, hashtags, and mentions that would not significantly contribute to sentiment analysis.
2. **Handling The Emoticons:** Cleaning emoticons from the dataset in sentiment analysis ensures a more focused analysis of the textual content, promotes standardization, reduces noise, and enhances compatibility with text-based sentiment analysis models. To handle this, a specific pre-processing step was likely employed to appropriately manage emoticons. This could include replacing them with sentiment indicators or mapping them to corresponding textual representations that reflect their intended emotions. This can improve the accuracy and reliability of sentiment analysis results on Twitter and other social media platforms.
3. **Tokenization:** Another essential pre-processing step performed on the Twitter data. Tokenization involves breaking down the text into individual units, such as words or subwords, to enable more effective analysis. By segmenting the data into meaningful units, each unit could be processed independently, facilitating subsequent analysis.
4. **Lowercasing:** To ensure standardization and avoid treating the same word with different cases as distinct entities, the text was converted to lowercase during pre-processing. This step aimed to achieve consistency in word representation and ensure that words like ”happy” and ”Happy” were treated as identical words during sentiment analysis.
5. **Removal of Stop Words:** Stop words are commonly used words in a language that do not carry significant meaning or contribute much to the overall understanding of a sentence. Examples of stop words in English include ”and”, ”the”, ”is” etc. And in Hindi “hai”,“ki”,“ka” etc. Stop words were removed from the English text using a predefined tool provided by NLTK (Natural Language Toolkit). NLTK offers a set of pre-defined stop words for the English language, which were employed to filter out common and non-informative words. This step aimed to reduce noise and enhance the accuracy of sentiment analysis by focusing on more meaningful content words. For the Hindi data set, a separate set of stop words was predefined to cater to the specific language. By utilizing these language-specific stop words, the model ensured the removal of irrelevant words in Hindi text, enabling more effective sentiment analysis.

4.2. Model selection and training. After pre-processing, we select a pre-trained model trained for the sentiment prediction of social media text. We select ”cardiffnlp/twitter-xlm-roberta-base-sentiment,” model which demonstrate the effectiveness in capturing sentiment information from social media data. We fine-tune this model for our task and during fine-tuning, the model's parameters are adjusted to optimize its performance in accurately predicting sentiment in social media text. Once the training process becomes complete, the trained model is utilized to perform sentiment analysis on the pre-processed text. This involved feeding the pre-processed text into the model and leveraging its learned representations to predict sentiment labels associated with the text. To prepare the text for analysis, we employ the model's tokenizer to tokenize our pre-processed data. We then pass the tokenized text as input to the pre-trained model. The model then predicts the sentiment labels for the data instances of our collected dataset. After getting the labeled dataset, we train six different machine learning models *viz.* Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Support Vector Machine and Ensemble learning Model for the final sentiment prediction of the multilingual code-mixed hate speech tweets.

4.3. Evaluation and Performance Metrics. We calculate various performance metrics to evaluate the performance of the sentiment analysis models. These metrics include accuracy, precision, recall, and F1-score. These metrics served as essential indicators of the model's effectiveness in accurately classifying sentiment. Accuracy measures the overall correctness of sentiment predictions, while precision quantifies the proportion of correctly predicted positive or negative sentiments out of all positive or negative predictions, respectively. Recall gauges the proportion of correctly identified positive or negative sentiments out of all actual positive or negative instances. The F1-score, which combined precision and recall, provided a balanced measure of the model's performance. Equation 4.1, 4.2, 4.3 and 4.4 show the mathematical equations of the above discussed metrics.

$$Accuracy = \frac{Correctly\ Classified\ Instance}{Total\ Instance} \tag{4.1}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.2}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.3}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

Figure 4.1 shows the value of above discussed evaluation metrics. Figure 4.2 visually represents the accuracy of the ensemble model. Furthermore, we analyze the performance of different machine learning models based on these metrics values. We can conclude that ensemble model gives better performance compared to the traditional machine learning models such as Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression and Support Vector Machine. This comparison aims to determine the superiority of the ensemble learning model in multilingual code-mixed sentiment analysis tasks for hate speech data, showcasing its strengths and advantages over traditional classifiers.

In order to visually represent the classifier's performance, we plot a confusion matrix (shown in Figure 4.3) for the ensemble model. The confusion matrix provides a comprehensive view of the classification results by displaying the true positive, true negative, false positive, and false negative predictions. By examining the confusion matrix, patterns of mis-classifications and the overall performance of the model can be easily observed and analyzed.

4.4. Result Analysis. The `sentiment_analysis` function plays a pivotal role in performing sentiment analysis on an instance using the `sentiment_task` pipeline. With the input of an instance, this function leverages a sentiment analysis model to determine the sentiment associated with it. It is reasonable to assume that the function yields a sentiment label or category as the output, representing the sentiment polarity of the instance, such as positive, negative, or neutral. To offer a user-friendly interface for the sentiment analyzer, we design our own interface using the Gradio library[2]. Through the gr. Interface function, the interface is defined, specifying the input type as "text" and the output type as "text". Additionally, the interface is accompanied by a title and description to enhance user understanding and interaction. Figure 4.4 and 4.5 are the example of sentiment label predictions obtained by the developed interface.

5. Conclusion. The multilingual code-mixed sentimental analysis model showcased remarkable precision in forecasting the sentiment polarity of hate speech tweets in multiple languages, comprising low-resource languages. The adoption of cross-lingual pre-training and fine-tuning approaches empowered the model to efficiently grasp and transfer knowledge across languages, thereby leading to better performance as opposed to monolingual models. We have also trained six different machine learning models such as Naive Bayes,

[2]https://gradio.app/

Bernoulli Naive Bayes	Prescision	Recall	F1-Score	Support
Negative	0.66	0.79	0.72	752
Neutral	0.67	0.50	0.57	627
Positive	0.72	0.73	0.73	561

Original Naive Bayes	Prescision	Recall	F1-Score	Support
Negative	0.65	0.78	0.71	752
Neutral	0.69	0.41	0.51	627
Positive	0.75	0.76	0.70	561

Multinomial Naive Bayes	Prescision	Recall	F1-Score	Support
Negative	0.66	0.79	0.72	752
Neutral	0.68	0.45	0.54	627
Positive	0.68	0.76	0.72	561

Logistic Regression	Prescision	Recall	F1-Score	Support
Negative	0.77	0.71	0.74	752
Neutral	0.65	0.73	0.69	627
Positive	0.76	0.74	0.65	561

Support Vector Classifier	Prescision	Recall	F1-Score	Support
Negative	0.75	0.67	0.71	752
Neutral	0.62	0.74	0.67	627
Positive	0.78	0.70	0.73	561

Ensemble Learning	Prescision	Recall	F1-Score	Support
Negative	0.67	0.79	0.72	752
Neutral	0.69	0.52	0.59	627
Positive	0.71	0.75	0.73	561

Fig. 4.1: Performance Metrics for Various Classifiers

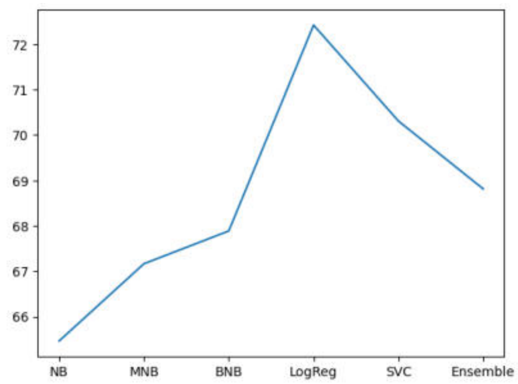

Fig. 4.2: Accuracy graph

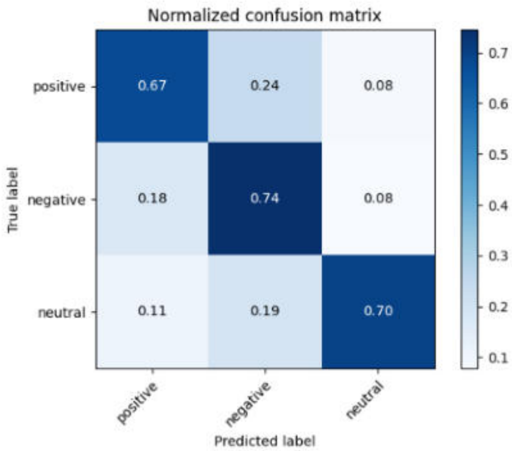

Fig. 4.3: Normalized confusion matrix for ensemble model

Fig. 4.4: Negative Classified Text

Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Support Vector Machine and Ensemble Learning Model. Among these six models, Ensemble learning model performs better compared to others. Despite this, the model has certain shortcomings and obstacles, such as struggles in detecting sarcasm and irony, and the necessity for significant computing resources. To enhance the performance of multilingual code-mixed sentiment analysis models, future research can address these difficulties.

6. Future Scope.

6.1. The occurring problem. Detecting hate and offensive content in social media is challenging due to linguistic complexity, and non-standard variations in grammar, spelling, and translation of language. While much work has been done on hate speech detection from tweets in English, the usage of regional languages in social media has increased the need for researchers to find hate speech from tweets. However, labeling tweets presents a challenge as a tweet can fall into multiple categories, and there may be subjective biases in labeling tweets. Programmed hate speech detection is also a closed system, and people try to circumvent the detection by posting content as images instead of text. To tackle this issue, optical character recognition can be used, but it is a constant battle between those spreading hateful content and those trying to block it. Furthermore, there are changing mentalities towards subjects over time and in historical context, which presents an additional challenge for programmed hate speech detection.

6.2. What can be done in the future to overcome the Problem. Future research can address the challenges faced by hate speech detection systems. One approach is to develop more accurate and unbiased labeling methods to minimize the subjective bias in labeling the text. Another solution is to incorporate context and historical information to understand the cultural and societal context of the text. Additionally, developing more robust models that can handle variations in language and user behavior can improve the performance of hate speech detection systems. Another solution is to integrate multiple sources of information, such as images

Fig. 4.5: Positive Classified Text

and user profiles, to gain a more comprehensive understanding of the content. Finally, developing systems that can adapt and learn from new data and changing user behavior can improve the long-term effectiveness of hate speech detection.

7. Limitation. This model requires a large amount of computing resources to train and use, which may be a limitation for some applications. The model may not perform well on datasets that are significantly different from the ones it was trained on, particularly if those datasets contain language or cultural nuances that are not well-represented in the training data. The model may struggle with tweets that use non-standard or informal language, such as slang or dialects until it is mentioned in the dataset. While the Multilingual Sentimental Analysis model outperforms other state-of-the-art models on several benchmark datasets, its performance may not be the best for every dataset and task, and it is always necessary to evaluate its performance on specific applications. We can avoid the resource limitations using pre-training concepts. We can also avoid the language bias limitations using transfer learning, by training various models on the datasets of numerous languages and combining those models to make an ensemble model for multilingual code-mixed sentiment analysis. We will explore these methods in our subsequent works.

REFERENCES

[1] M. M. Agüero-Torales, J. I. A. Salas, and A. G. López-Herrera, *Deep learning and multilingual sentiment analysis on social media data: An overview*, Applied Soft Computing, 107 (2021), p. 107373.

[2] S. Anbukkarasi and S. Varadhaganapathy, *Deep learning-based hate speech detection in code-mixed tamil text*, IETE Journal of Research, (2022), pp. 1–6.

[3] S. Banerjee, B. R. Chakravarthi, and J. P. McCrae, *Comparison of pretrained embeddings to identify hate speech in indian code-mixed text*, in 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), IEEE, 2020, pp. 21–25.

[4] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and 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.

[5] V. Gupta, V. Sehra, Y. R. Vardhan, et al., *Ensemble based hinglish hate speech detection*, in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2021, pp. 1800–1806.

[6] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, *Hate speech detection: Challenges and solutions*, PloS one, 14 (2019), p. e0221152.

[7] H. Madhu, S. Satapara, S. Modha, T. Mandl, and P. Majumder, *Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments*, Expert Systems with Applications, 215 (2023), p. 119342.

[8] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, *Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance*, Expert Systems with Applications, 161 (2020), p. 113725.

[9] S. K. Mohapatra, S. Prasad, D. K. Bebarta, T. K. Das, K. Srinivasan, and Y.-C. Hu, *Automatic hate speech detection in english-odia code mixed social media data using machine learning techniques*, Applied Sciences, 11 (2021), p. 8575.

[10] P. Rani, S. Suryawanshi, K. Goswami, B. R. Chakravarthi, T. Fransen, and J. P. McCrae, *A comparative study of different state-of-the-art hate speech detection methods in hindi-english code-mixed data*, in Proceedings of the second workshop on trolling, aggression and cyberbullying, 2020, pp. 42–48.

[11] T. Santosh and K. Aravind, *Hate speech detection in hindi-english code-mixed social media text*, in Proceedings of the ACM India joint international conference on data science and management of data, 2019, pp. 310–313.

[12] K. Sreelakshmi, B. Premjith, and K. Soman, *Detection of hate speech text in hindi-english code-mixed data*, Procedia Computer Science, 171 (2020), pp. 737–744.

[13] S. Thara and P. Poornachandran, *Code-mixing: A brief survey*, in 2018 International conference on advances in computing, communications and informatics (ICACCI), IEEE, 2018, pp. 2382–2388.
[14] E. Tromp, *Multilingual sentiment analysis on social media*, Master's Theisis. Department of Mathematics and Computer Science, Eindhoven University of Technology, (2011), pp. 1–104.
[15] W. Yin and A. Zubiaga, *Towards generalisable hate speech detection: a review on obstacles and solutions*, PeerJ Computer Science, 7 (2021), p. e598.

Edited by: Rajni Mohana
Special issue on: Sentiment Analysis and Affective computing in Multimedia Data on Social Network
Received: Jun 27, 2023
Accepted: Sep 26, 2023