TECHSSN at SemEval-2024 Task 10: LSTM-based Approach for Emotion Detection in Multilingual Code-Mixed Conversations

Ravindran V, Shreejith Babu G, Aashika Jetti
Rajalakshmi Sivanaiah, Angel Deborah S, Mirnalinee T T, Milton R S
Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering
Chennai - 603110, Tamil Nadu, India
{ravindran2213003, shreejithbabu2213006, aashika2210193}@ssn.edu.in,
{rajalakshmis, angeldeborahs, mirnalineett, miltonrs}@ssn.edu.in

Abstract

Emotion Recognition in Conversation (ERC) in the context of code-mixed Hindi-English interactions is a subtask addressed in SemEval-2024 as Task 10. We made our maiden attempt to solve the problem using natural language processing, machine learning and deep learning techniques, that perform well in properly assigning emotions to individual utterances from a predefined collection. The use of well-proven classifier such as Long Short Term Memory networks improve the model's efficacy than the BERT and Glove based models. However, difficulties develop in the subtle arena of emotion-flip reasoning in multi-party discussions, emphasizing the importance of specialized methodologies. Our findings shed light on the intricacies of emotion dynamics in code-mixed languages, pointing to potential areas for further research and refinement in multilingual understanding.

1 Introduction

The ultimate objective of this task EDiReF is to make progress in the field of conversational emotion recognition and reasoning. Analyzing emotions in natural language offers measurable understandings within the typically subjective domain of expressive language, connecting disciplines like psychology, cognition, and linguistics. This focuses on code-mixed Hindi-English dialogues, creating a unique and challenging linguistic setting in which participants must decipher the complexities of emotion recognition (ERC) and emotion flip reasoning (EFR). Code-mixing, or the intentional use of different languages within a single conversation, complicates the process and necessitates novel techniques for effective emotion recognition. This kind of collaborative work is critical because it reflects the changing environment of communication, where diversified language use needs complex models capable of understanding emotions in multilingual discussions resulting in overall development.

Our approach which focuses on the Emotion Recognition in Conversation (ERC) subtask, employs a sophisticated strategy based on deep learning methodologies suited to the intricacies of code-mixed Hindi-English (HI-EN) talks. At its core, our solution employs a Bidirectional Long Short-Term Memory (Bi-LSTM) network, an architecture capable of capturing intricate sequential connections inside text. We chose LSTM as LSTM networks are designed to overcome the limitations of traditional recurrent neural networks (RNNs) by mitigating the vanishing gradient problem and LSTM's ability to retain and selectively update information over time can lead to more accurate predictions of emotional states. We prioritized preprocessing, which includes tokenization and sequence padding, to help the model understand speech contexts. To enhance linguistic representation, the embedding layer is initialized with pre-trained word embeddings. For regularization and addressing overfitting issues, strategically placed dropout layers are incorporated. The training process utilizes categorical cross-entropy loss and the Adam optimizer.

The model showcased proficiency in precisely attributing emotions to individual utterances, showcasing its capability to decipher intricate emotional expressions in a multilingual context. From a quantitative standpoint, our system achieved an accuracy of 0.378 in sentiment analysis and secured the 23rd position among the competing teams. These findings not only provide insights into the model's strengths and areas for improvement but also highlight the importance of specialized mechanisms for complicated emotional connections in multilingual communication such as expanding datasets to encompass a broader range of linguistic and cultural contexts, as well as areas for future research

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 763–769

2 Background

The task at hand focuses on comprehending and categorizing emotions presented in code-mixed Hindi-English interactions. In this context, the input comprises dialogues where individuals communicate in both Hindi and English. The primary aim is to analyze each segment of these interactions and assign a precise emotion from a predefined set to capture nuanced sentiments. For example, throughout a conversation, people may display a variety of emotions at different points, such as joy, sorrow, or rage.

The task comprises two datasets centered around Multi-modal Sarcasm Detection and Humor Classification (MaSac) for sub-tasks 1 and 2, and Multimodal Emotion Lines (MELD) for subtask 3, but this paper majorly focuses on sub-task 1. The datasets for this task consist of code-mixed conversations, reflecting the real-world scenario where individuals seamlessly blend Hindi and English while communicating. The genre of the conversations may vary, encompassing diverse topics and contexts to ensure a comprehensive understanding of emotion dynamics in code-mixed language interactions. The size of the datasets is not explicitly mentioned, but it likely involves a substantial amount of annotated dialogues to train and evaluate emotion recognition models effectively. The dataset consisted of four parameters the episode name, the speakers list, the utterances, and the emotions mapped to the respective utterances. The emotions mapped to the utterances provided a standard for assessing models' performance in recognizing emotional expressions. For example, the utterance "ok, chalo rosesh chalo bahar" was mapped to the emotion "Contempt".

3 Related Work

Arora et al. (Arora et al., 2016) explores the capabilities of deep neural networks (DNN) using rectified linear units (ReLU). It introduces an algorithm for training a ReLU DNN with one hidden layer to global optimality with polynomial runtime in data size. The paper also improves lower bounds for approximating ReLU deep net functions and provides gap theorems for smoothly parametrized families of "hard" functions. Notably, it demonstrates the existence of functions requiring k^3 total nodes in a ReLU DNN with k^2 hidden layers, shedding light on the network's complexity.

Anshul Wadhawan and Akshita Aggarwal et al. (Wadhawan and Aggarwal, 2021) presents a Transformer-based approach for detecting emotions in code-mixed tweets. They introduce a Hinglish dataset, use bilingual word embeddings, and experiment with various models, including CNNs, LSTMs, and transformers like BERT. The BERT model achieves the best accuracy at 71.43%. The paper highlights the importance of emotion detection in social media and multilingual contexts, providing a valuable annotated dataset for future research.

Shivani Kumar et al. (Kumar et al., 2023a; Bedi et al., 2021) delved into Emotion Flip Reasoning (EFR) in multiparty conversations, showcasing state-of-the-art performance against baselines. Their research highlights the significance of EFR in enhancing empathetic response generation and understanding emotional dynamics in conversational settings, thus addressing the gap and providing insights into how specific remarks or expressions affect listeners.

Deepanshu Vijay et al. (Vijay et al., 2018) addresses emotion prediction in Hindi-English code-mixed social media text. They introduce a corpus from Twitter annotated with emotions and source languages. The paper proposes a supervised classification system using machine learning techniques and diverse features for emotion detection, contributing to resources for Hindi-English code-mixed text analysis in multilingual contexts.

In a parallel domain, a study focused on emotion analysis in low resource language Tamil is done by Varsini et al. (S et al., 2022). They have employed a lexicon-based approach and transformer models, utilizing dictionaries of words labeled with emotions. This research specifically addresses the challenges of extracting emotions from low resource texts in social media contexts, offering valuable insights.

Contextual emotion detection is executed using gaussian model and ensemble model by Angel Deborah et al. (Deborah et al., 2022; Angel Deborah et al., 2020). The challenge of contextual emotion detection in natural language processing has been addressed, emphasizing the difficulty for both machines and humans to accurately detect emotions like sadness or disgust in a sentence without sufficient context. The study underscores the growing importance of providing sensible responses in text messaging applications, where digital agents play a prominent role. The research showcases the efficacy of a Gaussian process for detecting contextual

emotions within sentences, comparing its performance with Decision Tree and ensemble models, including Random Forest, AdaBoost, and Gradient Boost.

Emotion recognition in Hindi-English code-mixed data, as explored in relevant papers, employs models like BERT, RoBERTa, CNNs, and LSTMs. The challenges highlighted align with our work, emphasizing the importance of addressing code-mixing complexities and the scarcity of annotated datasets. Similarly, in corpus creation for emotion prediction, the focus on a Twitter-based annotated corpus resonates with our efforts. The shared emphasis on overcoming linguistic diversity and cultural nuances underscores the mutual pursuit of enhanced accuracy in emotion recognition, urging continued research in these aspects.

4 System Overview

To optimise efficiency, we methodically integrated numerous critical algorithms and modelling decisions into our sentiment analysis model.

4.1 Data Preprocessing

4.1.1 Text Cleaning and Tokenization

The initial phase of our sentiment analysis model required thorough dataset preprocessing. The dialogue data (Kumar et al., 2023b, 2024) includes annotations for various emotions expressed by the speakers. These utterances are in both Hindi and English. We used Python's regular expressions and popular natural language processing packages to apply text cleaning techniques. Special characters, numerals, and unnecessary spaces were deleted. The cleaned text was then tokenized with TensorFlow Keras and (Arora, 2020) Indic NLP packages as It provides language-specific tokenization and other preprocessing functionalities tailored to languages spoken in the Indian subcontinent, improving the model's understanding of linguistic nuances.

4.1.2 Language-specific Tokenization

The dataset's multilingual composition necessitated the use of a complex tokenization technique. English utterances were tokenized with (Loper and Bird, 2002) NLTK's word tokenizer which breaks the text into individual words while preserving English language semantics, whereas Hindi text was tokenized with the Indic NLP package. It generates tokens by separating the text into its constituent words or tokens based on the identified boundaries. The goal of this multilingual tokenization technique was to identify and record language-specific patterns that were present across the dataset. The model is able to absorb and comprehend the unique linguistic aspects of both languages in the dataset because of this bilingual tokenization technique.

4.1.3 Stop Word Removal

In order to enhance the model's attention towards meaningful content, we systematically removed stopwords from both Hindi and English text. This important preprocessing step allowed for a more detailed understanding of the underlying sentiment by removing unnecessary noise and refining the raw data. In addition, we consistently converted all text to lowercase for uniformity and better generalization. This approach to preprocessing contributes to the model's robust performance in capturing the intricacies of emotion in code-mixed interactions.

4.1.4 Data Splitting:

The preprocessed data was divided into two parts: the training and the testing sets using the `train_test_split` function from the `scikit-learn` library. This ensures that the model's performance can be evaluated on unseen data, facilitating a thorough assessment of its generalization ability. In this approach, we could closely examine the extent to which our model could process novel, unseen data and see whether it could apply the knowledge it gained to a wider context.

4.2 Model Architecture

4.2.1 Embedding Layer

At the center of our sentiment analysis model is the Embedding layer. It takes tokenized words and transforms them into smart vectors. This layer, configured with an input dimension of `max_words`, an output dimension of 128, and an input length of `max_len`, converts tokenized input sequences into dense vectors that capture semantic associations between words. This layer essentially helps the model understand the deep connections and meanings between words in the input sequences.

4.2.2 Bidirectional LSTM

To capture the sequential dependencies in language effectively, we incorporated a Bidirectional Long Short-Term Memory (LSTM) layer (Staudemeyer and Morris, 2019). With 64 units, this bidirectional architecture facilitates the model in understanding

contextual relationships in both forward and backward directions.

4.2.3 Dense and Dropout Layers

A dense layer of 64 units was inserted sequentially, followed by Rectified Linear Unit (ReLU) activation (Arora et al., 2016). This layer, along with a dropout layer with a rate of 0.5, dramatically improved the model's ability to recognize complicated patterns while reducing overfitting. This is also adds a moderation in the learning process.

4.2.4 Output Layer

Using the softmax activation function (Sharma et al., 2017), the model's output layer successfully classified emotions into distinct categories as it converts the raw output scores of the model into probabilities, indicating the correct classification for the input. This categorical method enabled a more sophisticated comprehension of the diverse attitudes exhibited in the dataset.

In the final act, our model showcased its classification prowess through the output layer. With a touch of softmax activation function (Sharma et al., 2017), it skillfully categorized emotions into distinct categories. This categorical wizardry allowed our model to attain a nuanced understanding of the diverse attitudes presented in the dataset.

4.3 Model Training

4.3.1 Loss Function and Optimizer

The model was built using categorical cross entropy loss function and the Adam optimizer (Zhang and Sabuncu, 2018), known for its efficiency in handling sparse gradients. The rationale behind selecting the categorical crossentropy loss function and the Adam optimizer lies in their proven track record of effectiveness in sentiment analysis endeavors. Categorical cross-entropy performs well in circumstances with several classes, precisely meeting the requirements of sentiment classification with distinct emotion labels. By making the model allocate higher probabilities to the correct class, this loss function fosters more accurate sentiment predictions. This combination was intended to successfully optimize the model's weights, resulting in a robust learning process during training.

4.3.2 Training Parameters

A batch size of 32 and five epochs were used in the training procedure. This configuration produced the ideal training length by striking a balance between model convergence and processing efficiency.

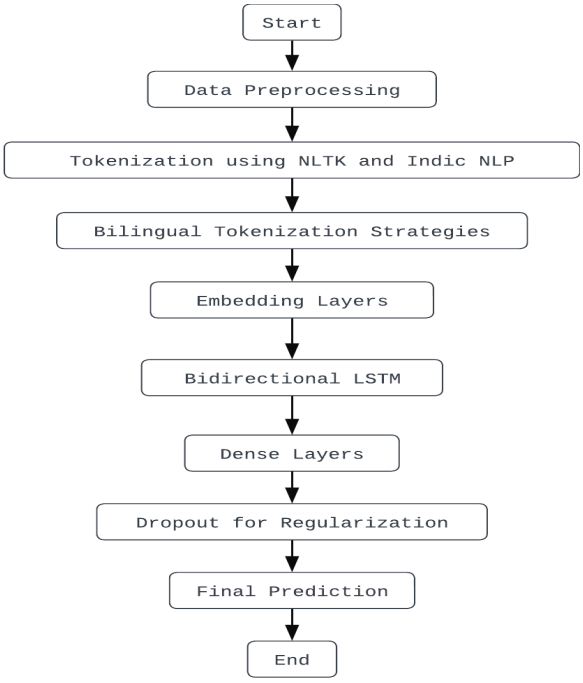

Figure 1: Model Process

4.4 Model Evaluation

4.4.1 Perfomance Metrics

The model's performance was evaluated using metrics including accuracy, loss, and weighted F1 score after training. These metrics provided a comprehensive understanding of how well the model identified emotions in textual input.

4.4.2 Prediction on the Test Set

Emotions on a test set were predicted using the trained model. This step consisted of applying the model to the preprocessed test data and decoding the predicted labels for further examination. The overall process of working is shown in Figure 1.

5 Experimental Setup

The data split used in the given code divides the dataset into training and testing sets. The split ratio is 80% for training data and 20% for testing data, as stated by the code's `test_size` argument of 0.2 to prevent overfitting and perform better on new data.

The learning rate is a crucial hyperparameter that determines the step size during the optimization

process.The learning rate, a key factor in the optimization process, was explored with values such as 0.01, 0.001, and 0.0001. To mitigate overfitting, dropout rates were varied, including options like 0.2 and 0.5

The number of LSTM units in the Bidirectional LSTM layer, a crucial aspect of model capacity, was adjusted with values like 32, 64, and 128. Additionally, the impact of different batch sizes (16, 32, 64) on convergence and computational efficiency was systematically explored.

In our experimental setup, we harnessed the power of scikit-learn for seamless implementation of various machine learning algorithms, handling data preprocessing tasks, and evaluating performance using diverse metrics. The NLTK library, an essential component, efficiently managed critical natural language processing functions, including tokenization, stopwords removal, and stemming.

To ensure model persistence and flexibility, we adopted joblib, a tool adept at saving and loading trained models. Moreover, our approach integrated external tools like NLTK Indic NLP, Scikit-Learn, and TensorFlow to elevate specific components of our sentiment analysis model. This encompassed optimizing tokenization, refining data splitting techniques, streamlining preprocessing steps, and conducting rigorous model evaluations, all contributing to the robustness and effectiveness of our experimental framework.

6 Results

The task is evaluated using the following performance metrics: precision, recall, accuracy and F1-score.

Recall indicates the classifier's ability to identify positive instances accurately and accuracy is defined as the ratio of the correctly predicted instances to the total number of instances in a dataset. It acts as a straightforward for the model's correctness while precision is a measure of how accurate the positive predictions are.

$$Precision = \frac{\text{TP}}{TP + \text{FP}} \quad (1)$$

$$Recall = \frac{\text{TP}}{TP + \text{FN}} \quad (2)$$

$$Accuracy = \frac{\text{TP + TN}}{TP + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

Approach	Accuracy
Word GloVe	0.35
Dist-Bert	0.30
LSTM Model	0.378

Table 1: Comparison of Accuracy for Different Approaches

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

$$Weighted\ F1\ Score = \sum_{i=1}^{N} W_i \cdot F1\ Score_i \quad (5)$$

The sentiment analysis model demonstrated the performance on the training dataset, achieving an accuracy of 0.39 and weighted F1 score of 0.38. The classification report is shown in Figure 2. On the test set, the model maintained the performance with an accuracy of approximately 0.378 and a weighted F1 score of 0.34 In the competition results, the model secured 23rd position. With LSTM approach, our model achieved an accuracy of 0.378 and secured the 23th position on the rankings. The comparison of various developed models are listed in Table 1.

We used a variety of approaches to improve the accuracy of sentiment analysis. One important approach was using pre-trained word embeddings, such as Word Glove (Rezaeinia et al., 2019), which captures semantic associations between words. The use of Word GloVe embeddings enabled a decent comprehension of contextual nuances, which contributed to increased sentiment analysis results with a F1 Score of about 0.35. While we initially explored the use of Word GloVe embeddings, we found that other methodologies yielded better results for our specific task. Therefore, we transitioned away from Word GloVe embeddings and pursued alternative approaches that demonstrated improved performance in managing the challenges associated with dual tokenization in mixed-language conversations. We also attempted to develop DistBERT, a transformer model, but faced a difficult case in which its integration resulted in an underwhelming F1 score of 0.30. This unexpected outcome spurred a rethinking of the implementation, prompting us to investigate alternate tactics and optimisations to improve the model's effectiveness.

	precision	recall	f1-score	support
anger	0.21	0.21	0.21	168
contempt	0.15	0.14	0.15	119
disgust	0.00	0.00	0.00	28
fear	0.14	0.11	0.12	83
joy	0.35	0.36	0.35	313
neutral	0.53	0.58	0.55	801
sadness	0.12	0.11	0.11	107
surprise	0.31	0.23	0.26	83
accuracy			0.39	1702
macro avg	0.23	0.22	0.22	1702
weighted avg	0.38	0.39	0.38	1702

Figure 2: Classification Report

Owing to the model's training on the restricted quantity of available data, it exhibits a bias towards specific emotions. "Okay chaliye dad, mein aapko bahar fenk kar aata hun!" is an example of a statement that should be predicted as "Joy," instead it is predicted as "neutral." The reason for the model's behaviour is that a lot of utterances are mapped to the neutral emotion; as a result, when a model is trained on this kind of data, it naturally becomes biased towards such emotion types.

In addition, a thorough analysis of the classification reports shed additional light on the theory on how class imbalances affect the model's functionality. As can be seen from the performance measures that were previously addressed, there are significant differences in the weighted and macro F1-scores, even if the classifiers' accuracy is the same for all datasets that were used. In particular, the sentiment analysis model performs noticeably better on the Hindi-English code-mixed dataset than on the Hindi and English monolingual datasets, highlighting the difficulties caused by the imbalances in the latter. This limitation is particularly notable for emotions with limited representation in the training data, emphasizing the need for strategies to address imbalances and improve the models' robustness across diverse linguistic scenarios.

7 Conclusion

We lay out a sentiment analysis system that can handle Hindi-English talks with mixed codes. Recognising and rationalising emotions in a bilingual environment was the main goal. The results show that the participants performed competitively in terms of emotion perception and reasoning, especially when there was frequent language change. The model using LSTM layers, NLTK Indic NLP provided the best result of 0.36 F1 score. Subtle emotional cues and particular code-mixing patterns continue to provide difficulties, nevertheless. Our method is noteworthy for its hybrid approach, which makes use of sentiment analysis, contextual embedding techniques, and language models that have already been trained and refined using code-mixed datasets.

It is still imperative to address specifics in low-resource languages in future development. Techniques like compiling lists of language-specific stop words have shown to be effective. Moreover, the effect of class disparities on the model's functionality is recognized. Furthermore, the effect of class imbalances on the performance of the model is recognized. Subsequent research could investigate customized approaches, including data enrichment or clustering techniques, to address these imbalances and improve the model's flexibility in a variety of language circumstances unique to our code.

References

S Angel Deborah, S Rajalakshmi, S Milton Rajendram, and TT Mirnalinee. 2020. Contextual emotion detection in text using ensemble learning. In *Emerging Trends in Computing and Expert Technology*, pages 1179–1186. Springer.

Gaurav Arora. 2020. inltk: Natural language toolkit for indic languages. *arXiv preprint arXiv:2009.12534*.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. 2016. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

S Angel Deborah, Rajendram S Milton, TT Mirnalinee, and S Rajalakshmi. 2022. Contextual emotion detection on text using gaussian process and tree based classifiers. *Intelligent Data Analysis*, 26(1):119–132.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023a. Emotion flip reasoning

in multiparty conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023b. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Seyed Mahdi Rezaeinia, Rouhollah Rahmani, Ali Ghodsi, and Hadi Veisi. 2019. Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139–147.

Varsini S, Kirthanna Rajan, Angel S, Rajalakshmi Sivanaiah, Sakaya Milton Rajendram, and Mirnalinee T T. 2022. Varsini_and_Kirthanna@DravidianLangTech-ACL2022-emotional analysis in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 165–169, Dublin, Ireland. Association for Computational Linguistics.

Sagar Sharma, Simone Sharma, and Anidhya Athaiya. 2017. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for hindi-english code-mixed social media text. In *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: student research workshop*, pages 128–135.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach. *arXiv preprint arXiv:2102.09943*.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.