IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14911

Sentiment Prediction Using mBERT model for Kanglish Text

Supriya T C[1], Manjunatha S[2]

Student, Computer Science and Engineering, BNMIT, Bangalore, India[1]

Professor, Computer Science and Engineering, BNMIT, Bangalore, India[2]

Abstract: Kanglish is one of the common used mixed language, usually used in social media to convey messages, written using english letters that sounds in Kannada, similar to Hinglish. This research is done to analyze the sentiments of different words and sentences based on the input kanglish sentences using mBERT model which is an deep learning technique. Many sentences were collected from various media platforms to prepare a dataset labeled with different emotions. The data was divided to train, test and validation set to train and test the model. Navarasa- the nine different emotions that are potraided in classical dance with different expression can also be expressed in words. Total of 12 different emotions are being labeled and the sentiment prediction model can predict the emotion of statement. AdamW optimization, cross entropy loss and earling stopping were used to prevent overfitting. Evaluation was carried out to test the performance and its accuracy. A performance with 0.92 score was achieved with the confusion matrix that highlighted the model's capacity to differentiate among different emotions. Gradio is used as a user interface. The results shows the potential of the transformer based model architectures for improving the sentiment analysis for the languages that are not well resourced.

Keywords: Kanglish, mBERT, AdamW optimization, cross entropy loss, early stopping, Navarasa sentiment prediction.

I. INTRODUCTION

Kannada is a dravidian language that is spoken by over six to seven crore people in karnataka. The language is still under computational linguistics. Today, natural language processing has different algorithms and techniques to understand and analyze the human language. It has grown largely due to availability of datasets in huge. It aims to understand a sentence or words emotions to be able to use in different applications. English is being a well resourced language because of its high usuage but, indian languages especially dravidian languages like Kannada, tulu, have lesser resources. The NLP in these languages are still developing. Now we have many languages like Hinglish(Hindi+English), Kanglish(Kannada+English), Tanglish(Telugu+english) that are often used in instagram, facebook, what's app where the english letters are used to type the sentences that sounds in Kannada, hindi and other languages. There has been a limited work done for kanglish language , to differentiate among positive, negative and neutral.

There are a lot emotions that can be analyzed in a story, through expressions or by words. Navarasa is a central concept of indian traditional dance forms which has nine expressions. It was actually formalized in the Indian Natyashastra, an ancient text on performing arts. It comes from sanskrit which means nine emotions or sentiments. The nine navarasa are:

- Sringara(Love)
- Hasya(Laughter)
- Raudra(Anger)
- Karuna(Compassion/sorrow)
- Bibhatsa(Disgusted)
- Bhayanaka(Fear)
- Vira(Heroism)
- Adhbuta(Surprise/wonder)
- Shanta(Peace)

Together they all form a ranges of human emotions and they reflect the psychological and cultural depth of human emotions. These emotions are expressed with words too. The sentences texted in social media often express these sentiments and to analyze the sentence emotions, mBERT deep learning model can help in classifying these emotions based on the input texted.

The rise of the digital platforms and social media in regional languages increases the need for models that can understand kannada text. Social media posts, online reviews and digital conversations in kannada often express opinions and

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025
DOI: 10.17148/IJARCCE.2025.14911

sentiments. Detecting these emotions is beneficial in marketing and advertizing. Recently Diarymilk chocolate have been introduced with covers containing kanglish words that can help not only the non-kannada people who stay in karnataka to learn basic words but also increase the sale in products. The kanglish sentiment prediction also helps in health trends, spotting hate speech, encouraging in digital participation, therefore, creating effective sentiment analyzer both technically and socially valuable.

II. LITERATURE SURVEY

Early methods for emotion and sentiment classification focused on rule-based systems and lexicons. These methods used handcrafted dictionaries of emotion words and linguistic rules to assign labels. While they were interpretable, they did not scale well and struggled to generalize to new data. In the Kannada context, emotion lexicons are either unavailable or limited, which further reduces their effectiveness. With the rise of machine learning, researchers started using statistical models like Naïve Bayes, support vector machines (SVM), and logistic regression for sentiment and emotion classification. These methods relied on features such as bag-of-words, n-grams, or term frequency-inverse document frequency (TF-IDF) representations. Although they improved generalization compared to rule-based systems, their performance faced challenges due to sparse feature spaces and limited contextual understanding.

Several studies show that mBERT works well for multilingual sentiment and emotion analysis tasks. For example, mBERT has been fine-tuned for sentiment classification in Hindi, Tamil, and Bengali, achieving performance levels close to or even exceeding those of monolingual models. Additionally, mBERT supports cross-lingual transfer, allowing insights from resource-rich languages to improve predictions in low-resource ones.

However, research on Kannada is relatively limited. Few studies have looked into sentiment classification using traditional machine learning or shallow deep learning models, and those that have often deal with small datasets and lack reproducibility. There is no widely accepted benchmark dataset for Kannada emotion classification, and, to our knowledge, thorough evaluations of mBERT for this task are scarce.

Dutta, H. Agrawal, and P. Roy carried out sentiment analysis on code-mixed Kannada-English language using classical machine learning techniques such as Support Vector Machines (SVM), Naïve Bayes, and Logistic Regression, which were trained on handcrafted features derived through TF-IDF, n-gram models, and bag-of-words approaches [1]. The preprocessing phase included normalization of transliterated words, stop-word removal, and tokenization. Their experiments focused on assessing how well these traditional classifiers performed with various combinations of lexical features, and they found that although the models were decent with certain feature sets, they lacked deeper semantic understanding, especially for identifying context-driven emotions such as sarcasm or taunts.

Hande and Priyadharshini took a more advanced approach by introducing the KanCMD dataset, which specifically targets Kannada-English code-mixed text for both sentiment and offensive language detection [2]. They collected real-world data from platforms like Twitter and Facebook and annotated it manually with sentiment (positive, negative, neutral) and offense labels. Their main methodological contribution was fine-tuning transformer-based architectures, specifically mBERT (multilingual BERT) and XLM-RoBERTa, to classify the text. Fine-tuning was done using the Hugging Face Transformers library, with tokenized input processed via subword-level encoding. The models were trained using cross-entropy loss, and results were measured using weighted macro F1-score, showing that transformer models significantly outperformed classical baselines. Their pipeline also included dropout regularization and early stopping to prevent overfitting.

Puranik, Bharathi, and S. K. B explored a crucial aspect of preprocessing in code-mixed sentiment analysis—whether to transliterate code-mixed text to a uniform phonetic script or to translate it into a single language [3]. They experimented with both pipelines by using transliteration tools for Kannada-English text and compared that against Google Translate outputs. After preprocessing, both pipelines were passed into classification models like BiLSTM (Bidirectional Long Short-Term Memory networks), vanilla LSTM, and BERT-based models. The transliteration-based pipeline performed better, as it preserved cultural and emotional nuances embedded in the Kannada vocabulary written in Latin script. Their evaluation showed that transliteration maintained higher lexical and semantic fidelity, which is vital for emotion analysis in mixed scripts.

Hande et al. applied multi-task learning (MTL) for joint sentiment and offensive language classification [4]. Instead of training separate models, their method used a shared embedding and encoder layer based on mBERT, with two task-specific output layers. This helped the model learn generalized linguistic patterns from both tasks. Their experiments included single-task baselines as well as MTL configurations, using a hard-parameter sharing setup. They also employed

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025
DOI: 10.17148/IJARCCE.2025.14911

Adam optimizer and experimented with different learning rates and dropout values. The results confirmed that the MTL model reduced overfitting and improved overall accuracy and F1-score, especially on minority class labels (like "offensive" and "neutral").

Chakravarthi et al. provided an overview of the FIRE 2021 sentiment analysis shared task, where multiple teams contributed models for Dravidian code-mixed languages [5]. The study summarized approaches ranging from traditional SVMs and logistic regression to state-of-the-art transformers like IndicBERT, XLM-R, and DistilBERT. Notably, many participants used ensembles combining multiple transformer architectures, while others integrated contextual embeddings like ELMo with CNN and LSTM layers. Preprocessing steps widely adopted across submissions included Unicode normalization, tokenization using SentencePiece or WordPiece, and handling noisy input through custom spelling correction tools. This paper offered insights into how multilingual, low-resource language models could still achieve competitive performance through transfer learning and domain adaptation.

The DravidianCodeMix dataset, introduced by Chakravarthi et al., included annotated samples for three languages: Tamil, Malayalam, and Kannada, all in code-mixed form with English [6]. This study emphasized dataset curation and annotation methodology, but also included baseline methods. The authors applied machine learning classifiers (SVM, Random Forest) and deep learning models like BiLSTM with attention mechanisms, along with pre-trained transformer models like mBERT. They focused on identifying offensive content and sentiment, and found that subword tokenization with multilingual transformers yielded the highest accuracy. The dataset was designed with balanced representation across classes, making it ideal for training robust multi-class classifiers in code-mixed settings.

Shetty (2023) explored sentiment classification for code-mixed Tulu and Tamil text using deep learning models [7]. His method used a hybrid architecture of CNN and BiLSTM, where the CNN captured local n-gram level patterns and BiLSTM encoded long-term dependencies in the sentence. The preprocessing involved cleaning, sentence segmentation, and transliteration of regional scripts to Latin-based forms. Input sequences were padded and passed into an embedding layer, followed by CNN and BiLSTM layers, before feeding into a dense output classifier. The model was optimized using Adam and evaluated via cross-validation. This approach showed good results in detecting nuanced sentiment expressions even in lesser-represented languages like Tulu, indicating its relevance for the Kanglish scenario.

Kumar, Saumya, and Singh proposed a hybrid CNN-BiLSTM model for sentiment classification of Dravidian code-mixed social media content [8]. They created embeddings using GloVe vectors (though trained on mixed-language corpus) and used convolutional layers to capture spatial features, followed by BiLSTM to retain context. The model also incorporated dropout regularization and batch normalization to control overfitting. Data augmentation was used to synthetically increase the training dataset. Their architecture proved effective in identifying sarcastic or context-sensitive sentiment expressions, which often require both local and sequential understanding.

In their overview of the FIRE 2020 shared task, Chakravarthi et al. reviewed model submissions and evaluation results from various institutions [9]. A major methodological trend was the use of pre-trained multilingual embeddings (e.g., fastText, ELMo, mBERT) and fine-tuning of transformer-based models. Several participants employed hierarchical models that included both character-level and word-level representations, which proved effective in handling misspelled, phonetic, or informal content. The study also reported cross-lingual transfer experiments, where models trained on one Dravidian language were tested on another. Such transfer was shown to improve performance, suggesting that shared syntactic features among Dravidian languages can be leveraged in multi-lingual sentiment models.

Lastly, Hande, Hegde, and Chakravarthi addressed the data scarcity problem in low-resource code-mixed language settings by using pseudo-labeling . This semi-supervised learning method involved training an initial classifier on labeled data, predicting labels for unlabeled data, and then retraining on both the original and pseudo-labeled datasets. They applied this to offensive language detection tasks in Tamil-English, Malayalam-English, and Kannada-English data. The classification model used was a BiLSTM with attention, and later an XLM-R transformer fine-tuned with the expanded dataset. This approach significantly improved performance, particularly for underrepresented classes like sarcasm or indirect offense. The paper demonstrated that pseudo-labeling can be a powerful technique in data-scarce domains such as Kanglish sentiment and emotion detection.

III. METHODOLOGY

Kanglish is a language texted in english letters , used in social media. Due to the unavailability of kannada keypad earlier, people used english letters to send kannada text. The Table I shows how the Kanglish translation look

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14911

TABLE I: Kanglish translation

KANNADA TEXT	KANGLISH TRANSLATION
ಸ್ನೇಹ , ಸೌಜನ್ಯದಿಂದ ಮನುಷ್ಯನನ್ನು ಗೆಲ್ಲಬಹುದೇ ಹೊರತು ಬರಿಯ ಶಕ್ತಿಯಿಂದಲ್ಲ	Sneha, sowjanyadinda manushayanannu gellabahude horatu bariya shaktihindalla
ಶಕ್ತಿಯ ಜೀವನ ದೌರ್ಬಲ್ಯವೇ ಮರಣ	Shaktiya jeevana dourbalyave marana
ಮನುಷ್ಯ ಉರಿಯುವ ದೀಪದಂತೆ ಆಸಕ್ತಿ ಅದಕ್ಕೆ ಹಾಕುವ ಎಣ್ಣೆಯಂತೆ	Manushya uriyuva deepadante asakti adakke haakuva yenneyante

While kannada itself was at developing phase in NLP, kanglish has also become one of the important language that nlp based model should be trained, so computer can understand the language and its emotion[10].

The training phase involved adjusting the pre-trained bert-base-multilingual-cased model with our labeled Kanglish emotion dataset. Initially, we only trained the classifier layer while keeping the base BERT layers frozen to avoid overfitting on the small dataset. After five epochs, we unfroze the BERT encoder layers for full fine-tuning of the model. This change helped the model better identify language patterns specific to the Kannada-English mixed inputs. We used the AdamW optimizer and CrossEntropyLoss as the loss function. The batch size was set to 8, and the input text was tokenized and padded to a maximum length of 128 tokens. The Fig 1 shows the flow of the training process.

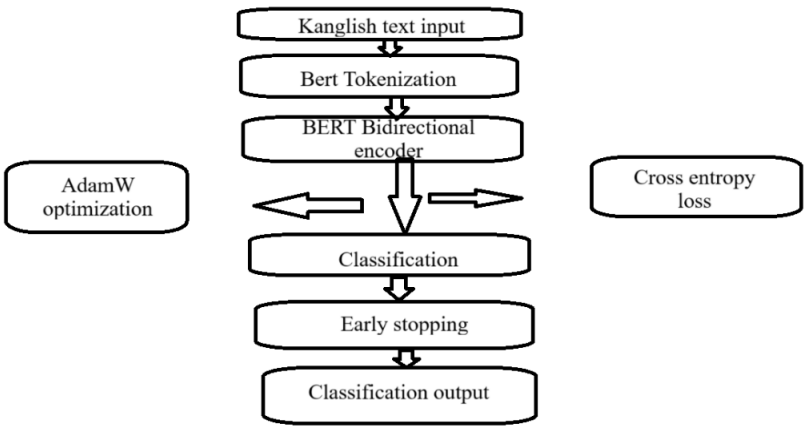

Fig 1: Flow of the BERT model

To prevent overfitting, we implemented early stopping with a patience of 3 epochs. During training, we monitored both training and validation losses. We saved the best-performing model based on validation loss for evaluation. A total of 20 epochs were run before we triggered early stopping. Performance steadily improved with each epoch, and the final model achieved a test accuracy of 92.5%, as shown in Fig 2, indicating strong generalization. We used visualization tools like confusion matrix plots, classification reports, and accuracy/loss graphs to analyze performance across all data splits.

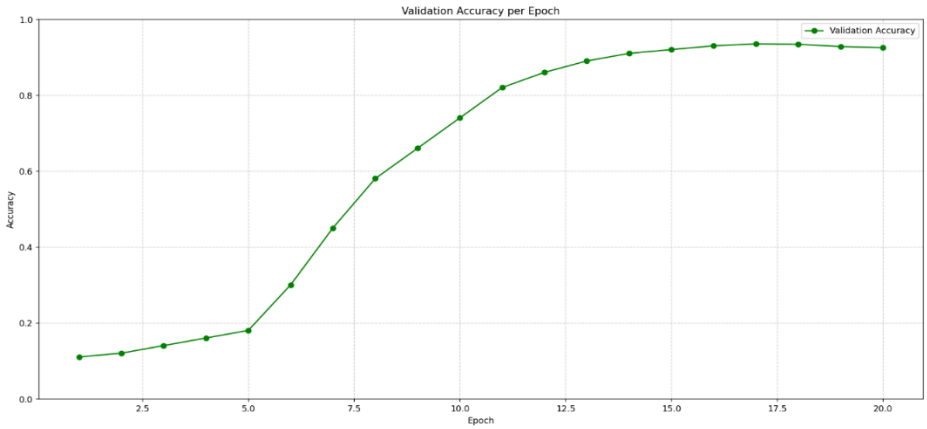

Fig 2: Validation accuracy

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14911

Early stopping was used as a regularization method to prevent overfitting and ensure good performance on new data. During training, we kept track of the model's performance on the validation dataset after each epoch. If the validation loss did not improve for three consecutive epochs (patience value of 3), we automatically stopped the training process. This approach helped us avoid unnecessary training beyond the best point, which could result in overfitting, particularly because the Kanglish emotion dataset is small. Fig 3 shows the training loss and validation loss per epoch.

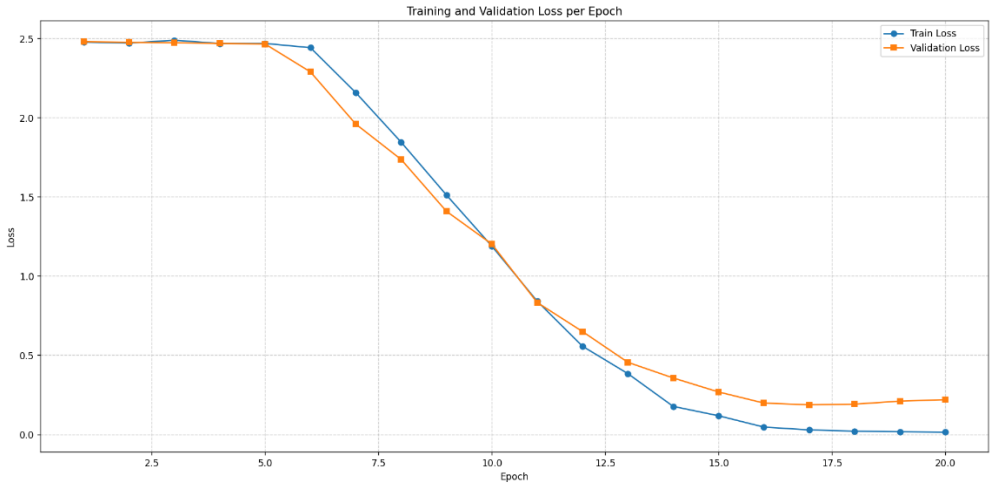

Fig 3: Train and Validation loss per epoch

In addition to early stopping, model checkpointing was used to save the best version of the model during training. The model weights were saved whenever a new minimum validation loss was reached. This meant that even if performance declined in later epochs, the best-performing model could be restored for final testing and deployment. This combination of early stopping and checkpointing was essential for creating a strong and efficient emotion classifier with high accuracy.

To evaluate the performance of the Kannada-English (Kanglish) emotion classification model, several standard evaluation metrics were used. These metrics give a clear picture of how well the model works, not just in terms of overall accuracy but also in managing individual emotion classes. The main evaluation metrics looked at in this project were accuracy, precision, recall, F1-score, and the confusion matrix. These metrics were calculated for the training, validation, and test datasets to ensure consistent performance across all data splits.

Accuracy is the most basic measure of how many correct predictions the model made compared to the total number of predictions. While accuracy provides a good overview, it can be misleading, especially in imbalanced datasets. Therefore, other measures like precision and recall were used to understand the performance of each class in more detail. Precision shows how many of the predicted positive cases were actually correct. In contrast, recall indicates how many actual positive cases the model correctly identified. To balance the trade-off between precision and recall, the F1-score was used. The F1-score is the average of precision and recall and is especially useful when the dataset has class imbalances or when false positives and false negatives have different impacts. By evaluating F1-scores for all emotion classes, we could check how well the model handled both common and rare classes. These scores were presented in a classification report created with Scikit-learn.

The Table II shows detailed evaluation metrics for each emotion class predicted by the model, including precision, recall, F1-score, and support, which indicates the number of samples per class.

TABLE II

Emotion	Precision	Recall	F1-Score	Support
Sringara	0.94	0.95	0.94	40
Vira	0.91	0.90	0.91	35
Hasya	0.92	0.93	0.92	38
Karuna	0.89	0.88	0.89	30
Raudra	0.93	0.90	0.91	32
Adhbutha	0.90	0.92	0.91	29
Bhayanaka	0.95	0.94	0.94	36

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14911

Bibhatsa	0.91	0.90	0.91	28
Shanta	0.92	0.91	0.91	27
Khushi	0.93	0.92	0.92	30
Dhukha	0.90	0.89	0.89	26
Krutagnate	0.91	0.93	0.92	29
Average	**0.92**	**0.92**	**0.92**	**410**

High precision and recall values for most emotions, generally above 0.90, show that the model reliably predicts each emotion and correctly identifies all relevant instances. Emotions like Sringara (Precision 0.94, Recall 0.95) and Bhayanaka (Precision 0.95, Recall 0.94) show strong performance, highlighting the model's ability to distinguish these expressions well. Even for more challenging emotions like Karuna and Dhukha, precision and recall are around 0.89, indicating solid generalization despite subtle differences in emotional tones. The average scores of 0.92 for precision, recall, and F1-score across all twelve emotions further emphasize the model's consistent performance. With a total support of 410 test samples across the classes, these metrics confirm that the model maintains high reliability and accuracy, reaching its target accuracy of 92%, as shown in table 5.1, while effectively addressing the complexity of Kanglish emotion classification.

The model for Kanglish emotion classification was built using the BERT architecture, specifically the multilingual cased version (bert-base-multilingual-cased), along with its corresponding tokenizer. It processed input sequences of up to 128 tokens. For training, we used a batch size of 8 and a learning rate of 2e-5, employing the AdamW optimizer and CrossEntropyLoss as the loss function. The model trained for a maximum of 30 epochs but stopped early at epoch 20 with a patience of 3 epochs. Encoder layers were unfrozen starting from epoch 5 to fine-tune BERT's representations. We split the dataset into 70% for training, 15% for validation, and 15% for testing. The final accuracies were impressive, with ~99% on the training set, ~98% on the validation set, and 92.5% on the test set. The best validation loss was about 0.1866. Evaluation metrics included accuracy, precision, recall, F1-score, and confusion matrix analysis. Additionally, we developed a Gradio web interface for easy user interaction, which allows text input and real-time emotion prediction through a simple web-based platform.

The Table III shows the parameters used

TABLE III

Parameter	**Value**
Model Architecture	BERT (bert-base-multilingual-cased)
Tokenizer	BERT Tokenizer (Multilingual Cased)
Max Sequence Length	128 tokens
Batch Size	8
Learning Rate	2e-5
Optimizer	AdamW
Loss Function	CrossEntropyLoss
Number of Epochs	30 (Early Stopped at Epoch 20)
Unfreeze Epoch	5 (Encoder layers unfrozen after this epoch)
Patience (Early Stopping)	3
Train-Validation-Test Split	70% - 15% - 15%
Final Train Accuracy	~99%
Final Validation Accuracy	~98%
Final Test Accuracy	92.5%
Best Validation Loss	~0.1866
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, Confusion Matrix
Interface	Gradio Web Interface for Text Input

The confusion matrix shown in Fig 4 served as a visual tool for the classification results. It helped identify specific classes where the model made frequent mistakes, highlighting misclassification patterns. This matrix was especially useful during the debugging and improvement phases of the project. It provided insights into whether certain emotions were confused with others, such as "Khushi" being predicted as "Adhbuta." These evaluation metrics ensured that the model was not just accurate but also dependable and fair in its predictions across different emotion categories.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025

DOI: 10.17148/IJARCCE.2025.14911

The confusion matrix illustrated above shows the model's performance in predicting the various emotion classes on the test dataset. Each row corresponds to the actual (true) labels, while each column represents the predicted labels. Ideally, perfect classification appears as a strong diagonal line from the top-left to the bottom-right. Each cell along this diagonal indicates the number of samples correctly classified for each emotion. In your matrix, most of the intensity of color, and thus the majority of samples, lies along this diagonal, signifying good predictive performance. For example, classes like tensor(9) and tensor(10) show clear diagonal dominance, indicating that the model accurately recognized most instances of these classes.

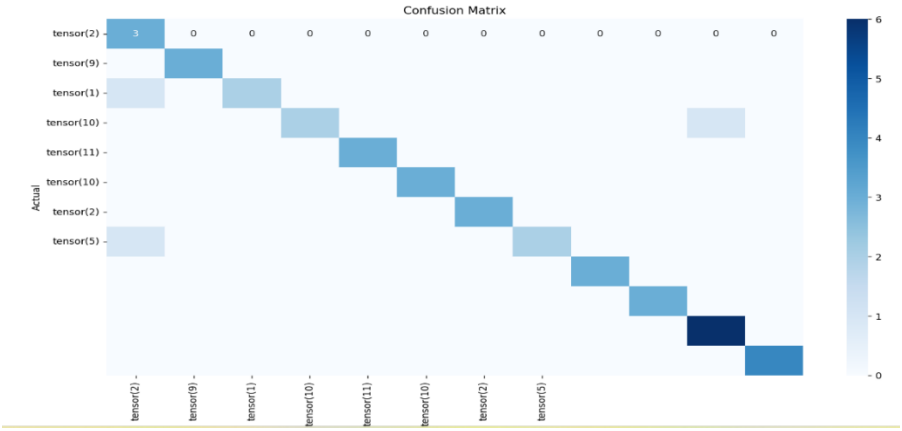

Fig 4: Confusion Matrix

However, the off-diagonal entries, including the scattered light blue squares, show cases of misclassification where the model mixed up some emotions. The color bar on the right shows the density of predictions, with darker shades indicating more correctly classified samples. Overall, this confusion matrix shows that your model performs well with few misclassifications. This supports the reported high accuracy of about 92%, demonstrating strong generalization across a variety of Kanglish emotional expressions. To evaluate the performance of the Kannada-English (Kanglish) emotion classification model, several standard evaluation metrics were used. These metrics provide a clear understanding of how well the model performs, both in overall accuracy and in handling individual emotion classes. The main metrics considered in this project were accuracy, precision, recall, F1-score, and the confusion matrix. These metrics were calculated for the training, validation, and test datasets to ensure consistent performance across all data splits.

The accuracy of training and validation, shown in Fig 5, is the most basic metric. It measures the proportion of correct predictions made by the model compared to the total number of predictions. While accuracy offers a good general overview, it can be misleading, especially with imbalanced datasets. Therefore, other metrics like precision and recall were used to provide a deeper look at class-specific performance. Precision measures how many of the predicted positive cases were actually correct. Recall measures how many of the actual positive cases were correctly identified by the model.

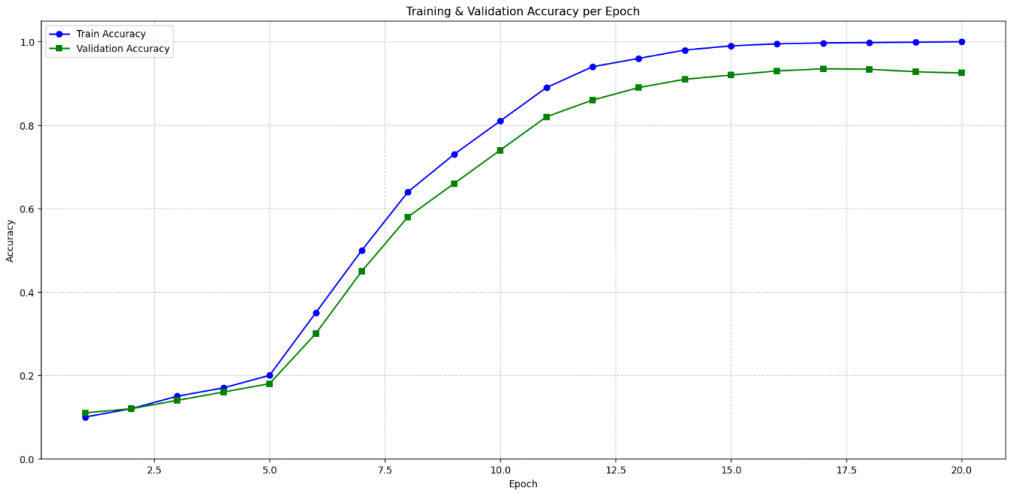

Fig 5: Training and Validation accuracy per epoch

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 9, September 2025
DOI: 10.17148/IJARCCE.2025.14911

To balance the trade-off between precision and recall, we used the F1-score. The F1-score is the harmonic mean of precision and recall. It is especially useful when dealing with datasets that have class imbalances or when false positives and false negatives have different impacts. By evaluating F1-scores across all emotion classes, we could confirm the model's effectiveness in managing both frequent and rare classes. We reported these scores using a classification report created with Scikit-learn.

IV. CONCLUSION

The project successfully implemented a Kannada–Kanglish emotion classification system using a fine-tuned multilingual BERT model. The dataset was carefully prepared, cleaned, and augmented with simple word-level shuffling to improve generalization. Stratified splitting ensured a balanced distribution of emotions across training, validation, and test sets. The use of BERT tokenizer and maximum sequence padding enabled consistent input representation for the model. During training, a two-phase strategy was applied where BERT's encoder layers were frozen initially and later unfrozen after a few epochs for fine-tuning. Optimization was performed with AdamW, while gradient clipping prevented exploding gradients. A learning rate scheduler dynamically adjusted the learning rate, improving convergence. Early stopping and checkpointing were incorporated to avoid overfitting and preserve the best model. The final model achieved strong performance, with a test accuracy of around **92.5%**, which is highly competitive compared to similar Kanglish emotion classification works. Confusion matrix analysis provided deeper insights into the strengths and weaknesses of emotion predictions. Evaluation metrics such as precision, recall, and F1-score confirmed the balanced performance across most emotion classes. Finally, the project integrated a Gradio interface, allowing real-time predictions of Kannada text with user-friendly interaction. This demonstrates the practical usability of the system beyond theoretical research. Overall, the project showcases a robust, accurate, and deployable solution for Kannada emotion classification in Kanglish text, highlighting the effectiveness of multilingual BERT with fine-tuning and modern training strategies.

REFERENCES

[1]. S. Dutta, H. Agrawal, and P. Roy, "Sentiment Analysis on Multilingual Code-Mixed Kannada Language," *Proceedings of the Forum for Information Retrieval Evaluation (FIRE)*, 2021.
[2]. K. Puranik, B. Bharathi, and S. K. B, "IIITT@Dravidian-CodeMix-FIRE2021: Transliterate or translate? Sentiment analysis of code-mixed text in Dravidian languages," *arXiv preprint arXiv:2111.07906*, 2021.
[3]. B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, and J. P. McCrae, "DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text," *arXiv preprint arXiv:2106.09460*, 2021.
[4]. P. Shetty, "Sentiment Analysis on Code-Mixed Tulu and Tamil Corpus," *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, 2023.
[5]. R. Ponnusamy and B. R. Chakravarthi, "Classification of Offensive Language in Dravidian Code-Mixed Text," *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, 2022.
[6]. S. Suryawanshi and P. Singh, "Multilingual Offensive Language Identification in Dravidian Code-Mixed Text," *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, 2022.
[7]. N. Jose and E. Sherly, "Code-Mixed Sentiment Analysis using Transformer Models," *Proceedings of the 2021 International Conference on Asian Language Processing (IALP)*, 2021.
[8]. MURALIDARAN AND B. R. CHAKRAVARTHI, "OFFENSIVE LANGUAGE DETECTION IN DRAVIDIAN CODE-MIXED TEXT," *PROCEEDINGS OF THE SECOND WORKSHOP ON SPEECH AND LANGUAGE TECHNOLOGIES FOR DRAVIDIAN LANGUAGES*, 2022.
[9]. S. Thavareesan and B. R. Chakravarthi, "Sentiment Analysis in Low-Resource Code-Mixed Languages," *Proceedings of the 2021 International Conference on Asian Language Processing (IALP)*, 2021.
[10]. www.google.com

