

Experiments with Sentiment Analysis of Code-mixed Tulu

H L Shashirekha, Rachana Nagaraju, B H Shekar
Department of Computer Science
Mangalore University
Mangalore, India

Abstract - Sentiment Analysis (SA), the task of automatically identifying polarity ('Positive', 'Negative', 'Neutral', or 'Mixed') in text, serves as a foundation for many applications. Tulu, a Dravidian language spoken in coastal Karnataka, India, remains a severely low-resource language with limited digital corpora and tools. In this paper, we investigate a spectrum of approaches ranging from traditional machine learning models trained with Term Frequency-Inverse Document Frequency features, to deep learning architectures, and transformer-based pre-trained multilingual models, to perform SA of Tulu text. Further, multiple ensemble strategies are experimented with soft voting, confidence-based selection, weighted probability averaging, and threshold-based methods, to improve the robustness. Tulu SA dataset of the shared task 'Sentiment Analysis in Tamil and Tulu' at DravidianLangTech@NAACL 2025 is used to build and evaluate the proposed models. Experimental results demonstrate that transformer-based approaches consistently outperform other baselines, achieving significant improvements in macro and weighted F1-scores on Tulu SA. Further, a comparative analysis of the models submitted to 'Sentiment Analysis in Tamil and Tulu' shared task illustrates that the proposed transformer-based model achieve competitive performance, often surpassing the other submissions in terms of macro F1-score.

Keywords - Sentiment Analysis; Low-resource Languages; Code-mixed Text; Social Media Analysis; Machine Learning; Deep Learning; Transformers.

I. INTRODUCTION

Sentiment Analysis (SA) is a core Natural Language Processing (NLP) task that aims to identify and classify opinions, emotions, or attitudes in textual content [1], [2]. With the ubiquity of digital platforms, SA has become essential in applications such as social media monitoring and customer feedback evaluation. Although significant advances have been achieved for high-resource languages like English, many low-resource Indian languages remain underrepresented due to the lack of annotated datasets, linguistic resources, and computational tools [3]. This challenge is particularly severe for Tulu - an extremely low - resource Dravidian language [4].

Tulu is a Dravidian language spoken primarily in the coastal districts of Dakshina Kannada and Udupi in the state of Karnataka and parts of Kasaragod in the state of Kerala, with an estimated 1.5 - 2 million speakers. Historically associated with the Tegalari script, contemporary Tulu content is predominantly

The authors are grateful to the Vision Group on Science and Technology (VGST), Department of Science and Technology, Government of Karnataka, for providing the financial support under the GRE scheme (GRD No: 1114) for carrying out this research work.

written in Kannada script. From an NLP perspective, Tulu is an extremely low-resource language presenting several challenges: i) scarcity of curated corpora and benchmarks, ii) lack of standardized orthography for Romanized Tulu and frequent spelling variation, iii) pervasive code-mixing with Kannada and English, iv) rich agglutinative morphology and dialectal variation, and v) lack of foundational tools (tokenizers, Part-Of-Speech taggers, morphological analyzers) and sentiment lexicons [3], [4]. These factors, compounded by small labeled datasets, make supervised modeling difficult and exacerbate domain shift across platforms.

SA of social media content introduces additional complexities: posts are short, noisy, and context-dependent that include creative spellings, elongations, emojis, hashtags, user mentions, and pragmatic phenomena such as sarcasm, irony, and implicit sentiment [1], [2]. A distinctive characteristic of code-mixing, where Tulu users interleave with other languages most prominently English and Kannada often with ad hoc romanization and transliteration. This results in script variation and informal orthography that complicate tokenization and feature extraction [5], [6]. Handling such code-mixed, multi-script text is therefore central to building robust, real-world SA systems for Tulu.

Cross-lingual learning has emerged as a critical paradigm in NLP, especially beneficial for low-resource languages. By transferring knowledge from well-resourced languages (e.g., English or Hindi), cross-lingual techniques can bootstrap effective NLP systems for underrepresented languages with minimal annotated data enabling better generalization and robustness in contexts where data scarcity hinders traditional model development [7]. This paradigm reduces the dependency on large annotated datasets and enables scalable development of NLP systems across diverse linguistic landscapes. Recent advancements in multilingual pre-trained models, such as mBERT and MuRIL, have further accelerated progress in cross-lingual transfer, demonstrating strong generalization capabilities even in zero-shot and few-shot settings. Many NLP applications including Machine Translation, Question Answering, Information Retrieval, Named Entity Recognition, Visual Question Answering, and SA, demonstrate the versatility and impact of cross-lingual learning in building inclusive and scalable systems.

To address data scarcity, cross-lingual learning leverages multilingual pre-trained models and annotated corpora from resource-rich languages to bootstrap models for low resource languages like Tulu. Further, cross-lingual representations help manage orthographic variation, share sub-word vocabularies across related languages, and enable models to generalize across

languages in zero-/few-shot settings. Recent shared tasks and community-led initiatives have further catalyzed progress in this domain. ‘Sentiment Analysis in Tamil and Tulu’^{1,2,3} shared tasks at DravidianLangTech⁴ workshops from 2023 to 2025 have introduced benchmark datasets and promoted model developments for SA in Tamil and Tulu [8, 9, 10]. Their efforts underscore both the challenges and promise of cross-lingual transfer using transformer-based architectures [11, 12].

To address the challenges of developing SA models for Tulu, in this paper, we describe a spectrum of approaches:

- Traditional Machine Learning (ML) models (Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Multilayer Perceptron (MLP)) trained with Term Frequency–Inverse Document Frequency (TF–IDF) of word unigrams and bigrams as features.
- Deep Learning (DL) architectures (RNNs, GRUs, LSTMs, CNNs, BiLSTMs, attention-based models).
- Transformer-based multilingual models (XLM-RoBERTa (XML-R), DistilBERT).
- Multiple ensemble strategies to enhance the robustness of the SA models.

The proposed approaches are built and evaluated using the official Tulu SA dataset released as part of the ‘Sentiment Analysis in Tamil and Tulu’ shared task at DravidianLangTech-2025. The dataset comprises user-generated, code-mixed Tulu text sourced from social media and annotated into five classes: ‘Positive’, ‘Negative’, ‘Neutral’, ‘Mixed’, and ‘Not-Tulu’. We follow the organizers’ train/dev/test splits in order to compare our proposed models with that of the shared task participants’ models. Table I presents code-mixed Tulu text and their English translations and the corresponding labels from the shared task dataset. Tulu code-mixed text highlight the diversity of sentiment categories and the prevalence of code-mixing in real-world data.

A comparative study of the proposed approaches illustrate that transformer-based models consistently outperform traditional ML baselines and DL architectures, achieving state-of-the-art results on Tulu SA. Further, a comparative study of our proposed models with those of the participants in the ‘Sentiment Analysis in Tamil and Tulu’ shared task at DravidianLangTech-2025 also illustrates the efficacy of our proposed models. This work thus highlights both the opportunities and challenges of applying transformer-based models to Tulu - an extremely low-resource Dravidian language and provide insights for building inclusive and scalable NLP systems.

The subsequent sections of this paper details the Related works (Section II), Methodology (Section III), Experiments, results, and implications of our approach (Section IV), followed by Conclusion and future works (Section V).

II. RELATED WORK

Recent years have seen growing interest in SA for low-resource Dravidian languages - Kannada, Tamil, Telugu,

Malayalam and Tulu, driven by advances in multilingual transformer-based models and the release of new annotated datasets. This line of research explores traditional ML and DL algorithms, cross-lingual transfer, hybrid architectures, and code-mixed text processing. Compared to high-resource languages, low-resource Dravidian languages face issues such as limited labeled data, high dialectal variation, and frequent code-switching between English and native scripts.

TABLE I. SAMPLE TULU TEXT WITH ENGLISH TRANSLATION AND CORRESPONDING LABEL

Tulu Text	English Translation	Label
ಏರೆನ್ ದೇವರ್ ಯೇದ್ ದಿವಾಡ್ ಮಾಡ್, ಮತ್ತೇ ನಿಕ್ ಏರೆನ್ ಜಾತ್ರೆಗ್ ಪೂವೊಲಿ	Once their temple festival rituals are over, you may go to their fair	Mixed
Vikas na kebit netter bathund	Vikas' ear was bleeding.	Negative
ಎನ್ನ ತುಳುವಪ್ಪೆ ಬುಲೊಂಡೊಲ್ವೆರ್ ತುಳುನಾಡ್ ಒಂಜಾಪಾನ್ ಬಲಿಪುನ ಅಶಿರ್ವಾದ್ ಪಡುನ ತೂ	Let us remember our Tulu forefathers' sacrifice for the unity of Tulu Nadu, and receive blessings	Neutral
That marla na getup make me laugh ...	That funny guy getup made me laugh	Positive
Ivattu concert super aayitu, friends jothe enjoy maadiddu	Today's concert was great, I enjoyed it with friends.	Not Tulu

The shared task on SA in Tamil and Tulu, organized as part of the DravidianLangTech workshop from 2023 to 2025 [8, 13, 14] aimed to promote SA in Tamil and Tulu - the low-resource Dravidian languages. Participants were provided with annotated datasets containing social media comments in Tamil and Tulu, labeled with three major sentiment categories: ‘Positive’, ‘Negative’, and ‘Neutral’, and other two categories: ‘Mixed’, and ‘Not-Tulu’. The task emphasized challenges such as code-mixed content, informal language, and multilingual scripts - particularly Tulu written in Kannada script. The shared task fostered innovation in sentiment classification and highlighted the importance of linguistic diversity in computational research. Various approaches ranging from traditional ML algorithms, DL architectures and transformer-based models like mBERT and IndicBERT, sentence embedding techniques such as SBERT and LaBSE, and hybrid frameworks combining neural networks with handcrafted features, were explored by the participants of these shared tasks.

Hegde et al. [11] developed one of the earliest annotated corpora for Tulu SA and applied traditional ML algorithms - LR and SVM, trained with TF-IDF features. Their results, achieving approximately 65 - 68% accuracy, established baseline metrics for code-mixed sentiment classification and highlighted the effect of feature sparsity on classifier generalization. Prabhu et al. [15] expanded the work on SA in Tulu by experimenting ensemble classifiers on the same corpus. They demonstrated that soft-voting ensembles combining SVM, RF, and Gradient Boosting, outperformed individual models by 3 - 4 F1-score. The merit of their approach was robustness across classes, but the ensemble still relied heavily on surface-level features, limiting cross-lingual generalization. Srichandra et al. [12] further enhanced SA in Tulu by applying cross-lingual learning employing back-translation to English and Kannada for text

¹ <https://codalab.lisn.upsaclay.fr/competitions/11095>

² <https://codalab.lisn.upsaclay.fr/competitions/16088>

³ <https://codalab.lisn.upsaclay.fr/competitions/20893>

⁴ <https://sites.google.com/view/dravidianlangtech-2025>

augmentation. This generated paraphrastic variants of Tulu sentences, improving robustness for underrepresented classes such as 'Mixed'. While effective, their approach depended on external machine translation quality and added computational overhead.

Hybrid and ensemble strategies that combine fine-tuned multilingual transformers with zero-shot or task-specific models (e.g., back-translation augmented models, and lexicon-guided components) have recently shown improved recall on minority classes such as 'Negative' and 'Mixed'. These hybrid approaches which borrow strengths from large multilingual encoders, lexical resources, and augmentation schemes represent the current state-of-the-art pathway for enabling robust NLP for severely low-resource and code-mixed languages like Tulu. Kundaragi and Joshi [16] explored zero-shot and few-shot transfer for Tulu SA using multilingual Natural Language Inference (NLI)-based models such as XLM-R. They showed that even without Tulu-specific training data, cross-lingual zero-shot transfer achieved over 50% macro F1- score, validating the potential of pre-trained multilingual encoders. However, their models struggled with domain adaptation to informal, code-mixed social media text.

More recent studies have specifically focused on low-resource and cross-lingual sentiment and emotion analysis. Koto et al. [17] proposed a lexicon-guided zero-shot sentiment pretraining approach, which significantly improves zero-shot performance across dozens of languages, including many low-resource and code-switched settings. Šmíd et al. [18] introduced constrained decoding for generative sequence-to-sequence cross-lingual aspect-based SA, achieving notable improvements for complex Aspect Based Sentiment Analysis (ABSA) tasks and further presenting a few-shot variant that delivers large gains with only a small number of target examples. Manafi and Krishnaswamy [19] examined cross-lingual transfer robustness under adversarial perturbations and demonstrated that transfer effectiveness is strongly influenced by linguistic relatedness and entity overlap between source and target languages as an important factor when selecting source languages for transferring to Tulu language. Wang et al. [20] applied knowledge distillation from monolingual to multilingual models to enhance both performance and interpretability in multilingual emotion detection, a technique that can also benefit SA in low-resource languages.

Several foundational works have established the methods and tools widely used for cross-lingual transfer. Ponti et al. [21] surveyed cross-lingual transfer methods, highlighting parameter sharing, multilingual embeddings, and adversarial learning as key strategies. Conneau et al. [22] introduced XLM-R, showing that large-scale multilingual pretraining substantially improves cross-lingual transfer for downstream tasks. Artetxe and Schwenk [23] proposed LASER for multilingual sentence embeddings, and Lample and Conneau [24] demonstrated the effectiveness of cross-lingual language model pretraining for zero-shot transfer. Benchmarks such as XTREME [25] and its extensions [26] formalized evaluation practices for multilingual generalization. While these studies laid the groundwork for cross-lingual and multilingual sentiment modeling, parallel

research has explored lightweight and hybrid systems that balance efficiency and contextual performance.

Collectively, these studies reveal that the combination of multilingual embeddings, contextual fine-tuning, and hybrid classifier ensembles has become the prevailing strategy for SA in Dravidian languages. Recent trends emphasize efficiency, adaptability to code-mixed settings, and integration of cross-lingual knowledge for improved robustness in low-resource environments. Moreover, augmentation and lexicon-guided pretraining continue to show promise in bridging the performance gap between high and low-resource languages.

III. METHODOLOGY

This section presents the pipelines developed for SA in code-mixed Tulu. The pipelines integrate specific pre-processing strategies, feature representation, model architectures, and training configurations. The methodologies span traditional ML models, DL architectures, and transformer-based architectures, reflecting a progressive exploration from traditional to state-of-the-art approaches.

A. Text Pre-processing

Social media text in code-mixed Tulu exhibits significant noise, transliteration inconsistencies, and heavy code-mixing with Kannada and English. To handle these challenges, the following pre-processing techniques are applied:

- **Noise Removal:** Removal of URLs, hashtags, user mentions, email addresses, digits, and redundant whitespace characters, using regular expressions. This reduces non-linguistic artifacts and stabilizes tokenization for both TF-IDF and DL models.
- **Emoji Handling:** Emojis are handled in two ways depending on the model pipeline. In classical and transformer-based pipelines, emojis are converted into textual descriptors using the emoji.demoji library⁵ preserving sentiment cues as tokens. In DL pipelines, emojis are removed entirely to reduce noise and simplify input representation.
- **Unicode Normalization:** Text is normalized with Unicode Normalization Form Compatibility Composition (NFKC)⁶ to canonicalize visually similar characters and reduce inconsistencies arising from transliteration and encoding variations.
- **Lexical Normalization:** Character elongations are reduced by limiting repeated letters to a maximum of two (e.g., gooooood → good), and common informal variations are standardized. This reduces feature sparsity while preserving the base semantic content of words.
- **Negation Handling:** Negation patterns are explicitly marked by inserting a token (e.g., [NOT]) before sentiment-bearing words (e.g., not good → NOT good). This ensures that polarity reversal is captured explicitly in feature-based and neural models.

⁵ <https://pypi.org/project/emoji/>

⁶ <https://unicode.org/reports/tr15/>

- **Sentiment Token Mapping:** Common sentiment-bearing words are replaced with standardized tokens such as [POS], [NEG], and [NEUT]. This helps models generalize sentiment patterns across different lexical variations.
- **Alphabetic Filtering:** In DL pipelines, non-alphabetic characters are removed, retaining only textual content. This simplifies the input space and improves sequence modeling for RNN-based architectures.
- **Case Normalization:** Romanized text is converted to lowercase to maintain consistency across different scripts and transliteration styles.
- **Invalid Entry Filtering:** Samples with missing values, empty text, or unmapped labels are removed to ensure data quality before training.

Together, these steps reduce orthographic noise, preserve sentiment signals (via tokenization and negation marking), and inject controlled diversity through back-translation. This pre-processing pipeline is applied consistently across traditional ML, DL, and transformer-based models to ensure that performance differences arise from modeling choices rather than inconsistencies in input representation.

B. Feature Representation and Model Building

We experimented with three broad families of learning models: i) traditional ML classifiers, ii) DL models, and iii) Transformer-based models. Each family is carefully designed to handle the low-resource and code-mixed characteristics of Tulu sentiment data.

1) *Traditional ML Models* - are trained with TF-IDF vectors of handcrafted word n-grams with $n = (1, 2)$ to capture local sentiment-bearing patterns (e.g., 'not good' is more informative as a bigram). Text data is high-dimensional by default and processing high-dimensional data is computationally expensive. Hence, rare terms with document frequency less than 2 and overly frequent terms with document frequency greater than 0.9 are discarded, and the number of features are restricted to 15,000. These features are vectorized using `TfidfVectorizer`⁷ and the vectors are used to train the following ML models:

- LR classifier is trained with balanced class weights and 'L₂' regularization ($C = 1.0$) and optimized with the 'lbfgs' solver for stable convergence on sparse high-dimensional features.
- RF classifier is configured with 200 trees and maximum depth of 20. Class balancing ensures minority classes such as 'Negative' and 'Mixed' are adequately represented. This model captures non-linear feature interactions that complement linear classifiers.
- SVM classifier is employed with a linear kernel, incorporating probability calibration via Platt scaling⁸ to generate soft labels for ensembling. Additionally, as the dataset considered for the experiments is imbalanced,

class weights are balanced to address class imbalance and mitigate data skew.

- MLP⁹ is a type of feedforward neural network that consists of multiple layers of neurons, typically organized into an input layer, one or more hidden layers, and an output layer. TF-IDF of word n-grams ($n = 1, 2$) up to a maximum 15,000 are used as input representations. Since TF-IDF produces high-dimensional sparse vectors, `StandardScaler` normalization is applied to stabilize training and improve convergence.

To evaluate the impact of network depth and model capacity, four MLP variants - MLP_Small, MLP_Medium, MLP_Large, and MLP_Deep are explored. These architectures are empirically designed by varying the number of hidden layers and neurons, rather than following predefined standard configurations. The hidden layer sizes range from shallow structures such as (100, 50) to deeper configurations like (300, 200, 100, 50), and an even deeper architecture with five layers (256, 128, 64, 32, 16) is also implemented.

To mitigate overfitting and improve generalization, several regularization techniques are employed, including 'ReLU' activation functions, dropout regularization in the range of 0.3 - 0.5, adaptive learning rate scheduling, and early stopping based on validation performance. These techniques are particularly important due to the high dimensionality of TF-IDF features and the relatively small dataset size.

These baselines provide a diverse modeling spectrum, balancing interpretability, efficiency, and expressive capacity. Collectively, they establish the strong baselines for SA in code-mixed Tulu text and effectiveness of using handcrafted features in low resource settings.

2) *DL Models* - are trained on word embeddings to capture sequential and compositional properties of Tulu text. To obtain low-dimensional dense word embeddings for Tulu text, randomly initialized embeddings of 100 dimensions are trained, enabling adaptation to code-mixed Tulu vocabulary. Recurrent architectures - RNN, GRU, LSTM, BiLSTM, and BiRNN, and CNN, are end-to-end neural architectures that learn feature representations by extracting embeddings from the input data. These embeddings are then used to train the models for SA. A brief description of these models is given below:

- RNN / GRU / LSTM - networks are used with hidden size 128, dropout 0.5, and sequence embeddings of size 100. GRU and LSTM capture long-term dependencies more efficiently than RNN.
- Bidirectional Models (BiRNN, BiLSTM) - process sequences in both forward and backward directions, improving the contextual capture in code-mixed text.
- CNN - Convolutional layers with kernel sizes (3, 4, and 5) and 100–150 filters are used to detect local n-gram

⁷ https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

⁸ <https://scikit-learn.org/stable/modules/calibration.html>

⁹ https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

sentiment patterns. A max-pooling layer is used to aggregate features before classification.

- Attention-Augmented Models - Self-Attention architecture is explored for BiLSTM and BiGRU models to learn token-level weights, focusing on sentiment-heavy words (e.g., ‘love’, ‘hate’) while ignoring noise.

All networks are optimized with Adam ($\text{lr} = 1e^{-3}$), trained for 25 epochs with batch size 32, and monitored with validation F1-score for early stopping. Each of these DL models are trained and evaluated independently to allow a direct comparison of individual model performances across different architectures.

3) *Transformer-based Models* - employ subword tokenization instead of word tokenization to overcome Out-Of-Vocabulary (OOV) problem. The multilingual transformer models - XLM-R, DistilBERT, and Zero-shot XLM-R learning, are implemented to leverage transfer learning:

- XLM-RoBERTa-base is originally trained on 100+ languages using massive CommonCrawl data. This gives it shared subword embeddings and contextual representations across languages. As Tulu is an extremely low-resource language and is not supported by any NLP tools or language models, XLM-R is fine-tuned with Multilingual Sentiment Analysis dataset for Indian languages¹⁰ and the given Tulu training data.

A class-weighted cross-entropy loss mitigates class imbalance. In addition to standard fine-tuning, a modified variant is explored where the default classification head is replaced with a deeper feedforward network consisting of multiple dense layers with dropout, enabling better modeling of complex decision boundaries and improving performance on minority classes.

- DistilBERT is a compact transformer model designed to bring the benefits of BERT-style architectures to multiple languages while being lighter and faster than full BERT or XLM-R. This model is fine-tuned with maximum sequence length of 512, trained jointly on Indic (Hindi, Kannada) and Tulu datasets to enhance cross-lingual transfer.
- Zero-shot XLM-R learning allows a model trained on one set of tasks or languages to generalize directly to new, unseen tasks or languages without any labeled data in the target domain. It relies on shared representations learned during pretraining, enabling knowledge transfer across domains. Zero-shot XLM-R is applied without fine-tuning by reformulating SA as a NLI problem. It reformulates sentiment labels as natural-language hypotheses (e.g. ‘The sentiment is positive’) and uses the Cross-lingual NLI (XNLI) framework to evaluate entailment between the input sentence (premise) and the hypothesis. Because XLM-R is pretrained on over 100 languages, it can transfer knowledge learned from high-resource languages directly to low-resource ones like Tulu. This demonstrates how multilingual embeddings and NLI reformulation enable effective SA in underrepresented,

code-mixed contexts without requiring any task-specific training data.

Together, these transformer-based models demonstrate complementary strategies for SA in code-mixed Tulu. Collectively, they highlight the power of multilingual embeddings and transfer learning for low-resource language tasks. The following configuration is used by the multilingual transformer models:

- Tokenizer Types: SentencePiece is used in XLM-R and WordPiece in DistilBERT. These subword tokenizers help to handle rare and unseen words effectively.
- Sequence Lengths: XLM-R uses length 128 while DistilBERT uses 512 tokens. This balances efficiency and contextual coverage.
- Shared Vocabulary: Multilingual vocabularies enable transfer learning across related languages. This is particularly beneficial for low-resource languages like Tulu.
- Class weight training: Applied to handle imbalance and improve performance in minority classes. In the modified XLM-R variant, explicit class weights are combined with a deeper classifier head to further improve representation learning for underrepresented classes.
- Zero-Shot Transfer Learning: Uses pre-trained multilingual knowledge without task-specific training. This shows the effectiveness of transfer learning in low-resource scenarios.

This configuration demonstrates the synergy between tokenizer design, sequence length, shared vocabularies, and training strategies in achieving robust SA for underrepresented languages like Tulu.

4) *Ensemble Strategies* - improve the robustness of the prediction by combining the outputs of independently trained models. In this work, ensembles are explored for traditional ML models, MLP-based models, and transformer-based models. The combination is performed after individual models generate their predictions or probability distributions.

- Traditional ML Ensemble: LR, RF and SVM are ensemble using soft voting strategy. Each classifier is trained independently on the TF-IDF features, and the final prediction is obtained by aggregating their class probabilities. This ensemble leverages the complementary strengths of linear, margin-based, and tree-based classifiers.

MLP Ensemble: MLP variants, namely MLP_Small, MLP_Medium, MLP_Large, and MLP_Deep, are ensemble using soft voting. Each MLP is trained independently on TF-IDF features followed by sparse-aware scaling, and their predicted probability distributions are aggregated to determine the final class label. This ensemble reduces variance and improves robustness across different network capacities.

¹⁰ https://huggingface.co/datasets/dhruv0808/indic_sentiment_analyzer

TABLE II. HYPERPARAMETERS FOR DIFFERENT MODEL FAMILIES

Model	Hyperparameters
LR	max_iter=1000, C=1.0, class_weight=balanced
RF	n_estimators=200, max_depth=20, class_weight=balanced
SVM	kernel=linear, C=1.0, probability=True, class_weight=balanced
MLP (Small/Medium/Large/Deep)	hidden=(100-300), layers=2-5, lr=0.001, early_stopping, max_iter=500 - 800
RNN/GRU/LSTM	hidden=128, dropout=0.5, lr=1e-3, epochs=25
BiRNN/BiLSTM	hidden=128, dropout=0.5, lr=1e-3, epochs=25
CNN	filters=100--150, kernel_sizes=(3,4,5), dropout=0.5, lr=1e-3
BiLSTM + SA	hidden=128, attention, dropout=0.5, lr=1e-3, epochs=25
BiGRU + SA	hidden=128, attention, dropout=0.5, lr=1e-3, epochs=25
XLM-Roberta	max_len=128, batch=8/, lr=2e-5, epochs=5--10, weight_decay=0.01
DistilBERT	max_len=512, batch=16, lr=2e-5, epochs=3, weight_decay=0.01

- Transformer-based Ensemble (Fine-tuned XLM-R + Zero-shot XLM-R learning): the independently generated class predictions along with their corresponding probability distributions over sentiment classes by both the models are combined at the prediction level using multiple ensemble strategies:
 - a. Voting: method prioritizes task-specific learning while maintaining agreement-based stability.
 - b. Confidence-based Selection: selects the prediction from the model with the higher maximum probability (confidence score). This method assumes that the model with higher confidence is more reliable for a given instance.
 - c. Weighted Averaging: combines probability distributions from both the models using fixed weights (0.7 for the fine-tuned model and 0.3 for the zero-shot model). The final prediction is obtained from the class with the highest combined probability. This allows controlled integration of both models while emphasizing the fine-tuned model.
 - d. Dynamic Combination: allows the prediction with higher confidence if both the models agree. But if they disagree, a weighted combination (0.8 fine-tuned and 0.2 zero-shot) is used to determine the final label. This strategy adapts based on agreement between the models.
 - e. Threshold-based Selection: allows the model prediction that predicts with confidence above a predefined threshold (0.8). Otherwise, a weighted combination of both models is used. This ensures that highly confident predictions are trusted directly.

Overall, these ensemble strategies improve robustness by combining complementary decision patterns from different models. While the traditional ML and MLP ensembles aggregate predictions from models trained on the same feature representation, the transformer-based ensemble combines task-specific fine-tuned knowledge with broader multilingual

generalization from zero-shot transfer learning. Table II summarizes the hyperparameters used in these models.

IV. EXPERIMENTS AND RESULTS

The objective of this study is to investigate the performance of various learning models under class imbalance while identifying the most effective approach for SA of low-resource code-mixed Tulu text. The experiments are conducted to evaluate the effectiveness of different pipelines for SA in code-mixed Tulu text.

A. Dataset Description

We employed two datasets and the description of the datasets is as follows:

- Tulu SA Dataset - released as part of ‘Sentiment Analysis in Tamil and Tulu’ shared task at DravidianLangTech-2025 [27] is used to build and evaluate the learning models for SA in code-mixed Tulu.
- Multilingual Sentiment Analysis Dataset for Indian Languages [28] is publicly available on Hugging Face¹¹. It consists of text in English and 11 Indian languages annotated with ‘Positive’, ‘Negative’, and ‘Neutral’ sentiment labels. This dataset is used to fine-tune XLM-R enabling effective cross-lingual generalization across diverse Indian languages.

Table III summarizes the statistics of these two datasets.

TABLE III. STATISTICS OF THE DATASETS

Attribute	Train	Validation	Test	Multilingual Dataset
Size	13,308	1,643	1,479	1,31,044
Null Values	7	1	0	52
Not Tulu	4,400	543	475	–
Positive	3,769	470	453	52,517
Neutral	3,175	368	343	32,520
Mixed	1,114	143	120	–
Negative	843	118	88	45,981

¹¹ https://huggingface.co/datasets/dhruv0808/indic_sentiment_analyzer

TABLE IV. PERFORMANCES OF THE PROPOSED MODELS (P – PRECISION, R – RECALL, F1 – F1-SCORE)

Model	P	R	F1	Accuracy
XLM-R	0.63	0.63	0.63	0.73
XLM-R (3-seed + Focal Loss + Upsampling)	0.64	0.59	0.61	0.70
XLM-R (Class-weighted + Deep Classifier)	0.59	0.59	0.58	0.67
XLM-R + Zero-Shot – Ensemble (Dynamic)	0.60	0.57	0.58	0.72
XLM-R + Zero-Shot – Ensemble (Voting)	0.58	0.57	0.57	0.72
XLM-R + Zero-Shot – Ensemble (Weighted)	0.58	0.56	0.56	0.71
DistilBERT	0.59	0.56	0.56	0.70
XLM-R + Zero-Shot – Ensemble (Threshold)	0.59	0.54	0.52	0.68
Logistic Regression	0.58	0.55	0.56	0.65
SVM	0.58	0.55	0.56	0.65
LR + RF + SVM – Ensemble	0.57	0.54	0.55	0.66
Random Forest	0.48	0.45	0.46	0.55
MLP Small	0.54	0.52	0.53	0.65
MLP – Ensemble	0.54	0.50	0.52	0.65
MLP Medium	0.53	0.49	0.50	0.65
MLP Deep	0.52	0.49	0.50	0.62
GRU	0.50	0.48	0.49	0.64
LSTM	0.50	0.48	0.49	0.63
BiLSTM	0.44	0.45	0.46	0.63
BiRNN	0.43	0.41	0.42	0.62
BiLSTM + Self-Attention	0.42	0.41	0.41	0.63
BiGRU + Self-Attention	0.41	0.39	0.40	0.62
CNN (ConvID)	0.40	0.39	0.40	0.63
RNN	0.40	0.39	0.40	0.63

B. Results of the Proposed Models

Since the class distributions are highly imbalanced, macro-averaged metrics (Precision, Recall, and F1-score) are chosen as primary evaluation measures in addition to overall accuracy. This ensured that performance on minority classes (e.g., ‘Negative’ and ‘Mixed’) was fairly considered alongside majority classes (e.g., ‘Neutral’ and ‘Positive’).

The performances of individual models across different families of architectures and ensembles are shown in Table IV. The results clearly indicate that transformer-based architectures substantially outperform traditional ML and DL baselines. Fine-tuned XLM-R achieved the highest overall performance with a macro F1-score of 0.63 and an accuracy of 73%, demonstrating its strong ability to capture contextual nuances in Tulu text. The 3-seed fine-tuned variant with focal loss and upsampling further showed improvements in class balance, though with slightly lower overall macro F1-score, suggesting a trade-off between robustness to minority classes and overall generalization.

Ensemble of Zero-shot transfer learning and fine-tuned XLM-R provided consistent improvements in stability, reaching macro F1-scores in the range of 0.56 - 0.58, with slightly higher

accuracy due to majority-class sensitivity. DistilBERT, despite being a lighter model, performed competitively (macro F1-score 0.56), highlighting the potential of distilled architectures for low-resource deployment scenarios.

Traditional ML models exhibited reasonable performance (macro F1-scores in the range 0.46 - 0.56), validating the utility of TF-IDF features as strong baselines. Similarly, MLPs achieved stable results (macro F1-score in the range 0.50 - 0.53), with shallow architectures slightly outperforming deeper counterparts.

Recurrent architectures (RNN, GRU, LSTM, BiLSTM) and their attention-enhanced extensions underperformed compared to transformer models, with macro F1-scores between 0.40 and 0.49. While these models captured some sequential dependencies, their limited ability to model long-range context and handle imbalanced data contributed to weaker results. CNN-based models demonstrated similar trends, performing on par with simple RNNs.

Overall, the results demonstrate that XLM-R outperformed as the most effective solution for SA in code-mixed Tulu. Traditional baselines and recurrent models provide valuable insights but fall short of transformer-based architectures in handling the complex linguistic characteristics and imbalanced nature of the dataset.

To contextualize our findings, we compare the performance of XLM-R - our best performing model - with the top performing models of ‘Sentiment Analysis in Tamil and Tulu’ shared task at DravidianLangTech@NAACL 2025 [29] as shown in Table V.

TABLE V. COMPARISON OF THE PERFORMANCE OF XLM-R - OUR BEST PERFORMING MODEL WITH TOP MODELS OF TULU SENTIMENT ANALYSIS SHARED TASK

Team Name	F1-Score
XLM-R (Our best performing model)	0.6300
lowes	0.5938
ET2025	0.5882
Hermes	0.5801
JustATalentedTeam	0.5617
SSNTrio	0.5609
Lemlem	0.5583
YenLP_CS	0.5511
codecrackers	0.5425
RMKMavericks	0.5318

C. Error Analysis

It can be observed that our fine-tuned XLM-R with a macro F1-score of 0.63 and accuracy of 73% surpasses the top-performing system and demonstrates stronger robustness across minority classes. This validates the effectiveness of our pipeline while confirming consistency with the competitive benchmark results reported in the shared task.

To investigate the causes of misclassifications, a detailed analysis was conducted using the confusion matrix as shown in Figure 1 and representative error samples are shown in Table VI. The confusion matrix provides a quantitative overview of

prediction behavior across classes, while the error examples offer qualitative insights into the types of linguistic and contextual patterns that lead to incorrect predictions. Together, these analyses reveal that the observed errors are systematic and arise from identifiable limitations in class separability, token dominance, and sentiment interpretation rather than random prediction noise.

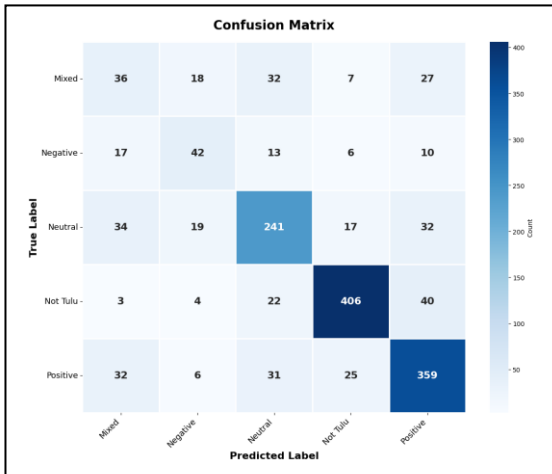


Fig. 1. Confusion Matrix of XLM-R Model

TABLE VI. SAMPLE MISCLASSIFIED INSTANCES

Tulu Text	English Translation	True Label	Predicted Label
ಇರೆನರ್ಡ್ ಸಂತುನ comedy ಸೂಪರ್ ಉಂಡು ಮಾರ್ ಸೆರಿಯಸ್ಲಿ	Santu's comedy is super	Negative	Positive
ಹೋರಾಟ ಹೋರಾಟ #ಸಂತು ಉಲಾಯಿ ಬರಮುಟ್ಟು ಹೋರಾಟ	Strike, strike — strike until Santu comes inside.	Neutral	Not Tulu
Anchoring malpunar er onthe jasthi tulu ne patherle.	Anchors, please use more Tulu while speaking.	Mixed	Negative
ರಾಗಿ ಮುದ್ದೆ ಮರ್ಯಾದೆ ಹೋಯಿತು	"The respect for ragi balls has faded"	Positive	Not Tulu
"Sama beary bundle, pinendre joke akro"	Speak Beary properly, don't joke	Not Tulu	Positive

1) Error Distribution and Structural Causes

The model achieves an overall accuracy of 73.29%, with 395 misclassifications out of 1,479 samples. However, the distribution of these errors is highly uneven across classes, indicating that the errors are driven by structural issues rather than random noise.

The 'Mixed' class shows the highest error rate (70.0%), followed by 'Negative' (52.3%). In contrast, 'Not Tulu' has the lowest error rate (14.5%), indicating that language identification is significantly easier than sentiment discrimination. This disparity reveals that the model is not uniformly learning all classes; instead, it is biased toward classes with stronger lexical signals.

2) Confusion-Driven Failure Modes

The confusion matrix reveals several dominant and repeated error pathways:

- 'Not Tulu' → 'Positive' (40 cases): This is not a random error. It occurs because the model prioritizes sentiment-bearing tokens over language identification. Words such as 'super', 'good', and similar positive expressions appear in both Tulu and non-Tulu sentences. Since the model is trained for sentiment classification, these tokens dominate the prediction, causing misclassification of language-specific labels.
- 'Neutral' ↔ 'Mixed' (34 and 32 cases): The confusion between 'Neutral' and 'Mixed' is bidirectional and symmetric, indicating that the model fails to learn a clear boundary between these classes. This is primarily because both classes lack strong polarity markers. 'Mixed' class samples often contain weak or implicit sentiment, which becomes indistinguishable from 'Neutral' representations in the embedding space.
- 'Positive' → 'Mixed' (32 cases) and 'Positive' → 'Neutral' (31 cases): These errors arise when positive sentiment is expressed without strong intensifiers. The model relies heavily on explicit sentiment words; when such words are absent or minimal, the representation shifts toward 'Neutral' or 'Mixed' regions.
- 'Mixed' → 'Positive' (27 cases): This indicates a systematic positive bias. When a sentence contains both positive and negative cues, the model tends to overweight the positive component, leading to incorrect classification.

3) Representation-Level Limitations

The observed errors are strongly tied to the way model encodes textual information:

- Dominance of High-Frequency Tokens: Frequently occurring tokens such as 'movie', 'video', 'comedy', and 'song' appear across multiple classes. These tokens carry little discriminative value but heavily influence the embeddings due to their frequency, leading to misclassification, especially toward 'Neutral' and 'Positive' classes.
- Weak Encoding of Subtle Negativity: Negative sentiment is often expressed implicitly rather than through strong negative words. As a result, the model fails to assign sufficient weight to these subtle cues, causing 'Negative' samples to shift toward 'Neutral' predictions.
- Overlapping Embedding Regions: 'Neutral', 'Mixed', and 'Positive' class samples occupy overlapping regions in the learned representation space. This overlap is reflected in the high confusion rates among these classes, indicating that the model does not learn sufficiently separable decision boundaries.

4) Input-Specific Failure Patterns

Analysis of misclassified samples reveals that certain input characteristics consistently lead to errors:

- Short Inputs - Very short texts (e.g., 2 - 3 words) do not provide enough contextual information for reliable

classification. These samples are frequently mapped to dominant classes such as ‘Neutral’ or ‘Not Tulu’.

- Repetitive Sentiment Words: Sentences containing repeated positive words (e.g., ‘super super’) disproportionately influence the model toward ‘Positive’ predictions, even when the true label differs.
- Code-mixed or Cross-lingual Text: The presence of multiple languages within a single sentence disrupts the model’s ability to correctly identify both sentiment and language, leading to confusion between ‘Not Tulu’ and other sentiment classes.
- Lack of Explicit Sentiment Indicators: Sentences that rely on contextual or implicit sentiment (rather than explicit words) are often misclassified, particularly into ‘Neutral’ or ‘Mixed’ categories.

5) Interpretation of Model Behavior

The error patterns indicate that the model:

- Relies heavily on explicit sentiment tokens rather than contextual understanding.
- Exhibits a positive prediction bias when conflicting signals are present.
- Struggles to model fine-grained sentiment distinctions, especially between ‘Neutral’ and ‘Mixed’.
- Prioritizes semantic similarity over class boundaries, leading to overlapping predictions.

These findings demonstrate that the misclassifications are systematic and arise from identifiable limitations in representation learning, class separability, and reliance on dominant lexical features, rather than random prediction errors.

V. CONCLUSION AND FUTURE WORK

This study explored SA in Tulu, a low-resource Dravidian language, tackling challenges of code-mixed text and limited annotated data. A spectrum of approaches ranging from traditional ML models trained with TF-IDF features, to DL architectures, and transformer-based pre-trained multilingual models, are investigated to perform SA of Tulu text. Further, multiple ensemble strategies are also experimented with soft voting, confidence-based selection, weighted probability averaging, and threshold-based methods, to improve the robustness. Experiments are conducted using the ‘Sentiment Analysis in Tamil and Tulu’ shared task dataset at Dravidian-LangTech@NAACL 2025 and the results are compared with that of models submitted by the participants' to this shared task. The experimental results demonstrate that XLM-R - a fine-tuned transformer-based multilingual model outperform other baselines, achieving the highest macro F1-score and accuracy. The study highlights the effectiveness of cross-lingual transfer learning and ensemble strategies to improve model robustness in low-resource scenarios. Future work includes experimenting different approaches for handling data imbalance, exploring various multilingual models and extending the approach to other low-resource Dravidian languages.

REFERENCES

- [1] B. Liu, “Sentiment Analysis and Opinion Mining,” ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012, vol. 5, no. 1.
- [2] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A Survey on Sentiment Analysis Methods, Applications, and Challenges,” *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.
- [3] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, “The State and Fate of Linguistic Diversity and Inclusion in the NLP World,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, 2020, pp. 6282–6293.
- [4] P. Shetty, “Natural Language Processing for Tulu: Challenges, Review and Future Scope,” in *Speech and Language Technologies for Low-Resource Languages*. Springer, 2024, pp. 93–109.
- [5] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, “I Am Borrowing Ya Mixing?” An Analysis of English-Hindi Code Mixing in Facebook,” in *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, 2014, pp. 116–126.
- [6] S. Alam, M. F. Ishmam, N. H. Alvee, M. S. Siddique, M. A. Hossain, and A. R. M. Kamal, “BnSentMix: A Diverse Bengali-English Code-Mixed Dataset for Sentiment Analysis,” in *Proceedings of the International Conference on Language Resources and Evaluation (LREC-COLING 2024)*. ELRA, 2024, pp. 11 089–11 098.
- [7] T. Adimulam, S. Chinta, and S. K. Pattanayak, “Cross-Lingual Learning Techniques for Low-Resource Languages: A Case Study,” in *Proceedings of the International Conference on Recent Advances in Engineering and Technology (ICRAET)*. ER Publications, 2025.
- [8] A. Hegde, B. R. Chakravarthi, H. L. Shashirekha, R. Ponnusamy, S. C. Navaneethakrishnan, L. S. Kumar, D. Thenmozhi, M. Karunakar, S. Sriram, and S. Aymen, “Findings of the Shared Task on Sentiment Analysis in Tamil and Tulu Code-mixed Text,” in *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*. Virtual Event: Association for Computational Linguistics, Mar. 2023, pp. 73–82.
- [9] L. S. Kumar, A. Hegde, B. R. Chakravarthi, H. L. Shashirekha, R. Natarajan, S. Thavareesan, R. Sakuntharaj, D. Thenmozhi, P. K. Kumaresan, and C. Rajakumar, “Overview of Second Shared Task on Sentiment Analysis in Code-mixed Tamil and Tulu,” in *Proceedings of the Fourth Workshop on Speech and Language Technologies for Dravidian Languages*. Gyeongju, Republic of Korea: Association for Computational Linguistics, Mar. 2024, pp. 75–84.
- [10] D. Thenmozhi, B. R. Chakravarthi, A. Hedge, H. L. Shashirekha, R. Natarajan, S. Thavareesan, R. Sakuntharaj, K. Kalyanasundaram, C. Rajkumar, P. Shetty, and H. S. Kumar, “Overview of the Shared Task on Sentiment Analysis in Tamil and Tulu,” in *Proceedings of the Fourth Workshop on Speech and Language Technologies for Dravidian Languages*. Virtual Event: Association for Computational Linguistics, Mar. 2025, pp. 865–872.
- [11] A. Hegde, M. D. Anusha, S. Coelho, H. L. Shashirekha, and B. R. Chakravarthi, “Corpus Creation for Sentiment Analysis in Code-Mixed Tulu Text,” in *Proceedings Of The 1st Annual Meeting of The ELRA/ISCA Special Interest Group On Under-Resourced Languages*. ELRA, 2022, pp. 33–40.
- [12] I. V. Srichandra, H. Vijay, P. O. Rao, and B. Premjith, “Deep Learning Approach for Sentiment Analysis in Tamil and Tulu – dravidian-langtech@naacl 2025,” in *Proceedings of The Fifth Workshop on Speech, Vision, And Language Technologies For Dravidian Languages*. ACL, 2025, pp. 45–54.
- [13] L. Sambath Kumar, A. Hegde, B. R. Chakravarthi, H. Shashirekha, R. Natarajan, S. Thavareesan, R. Sakuntharaj, T. Durairaj, P. K. Kumaresan, and C. Rajkumar, “Overview of Second Shared Task on Sentiment Analysis in Code-mixed Tamil and Tulu,” in *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, B. R. Chakravarthi, R. Priyadharshini, A. K. Madasamy, S. Thavareesan, E. Shery, R. Nadarajan, and M. Ravikiran, Eds. St. Julian’s, Malta: Association for Computational Linguistics, Mar. 2024, pp. 62–70.
- [14] D. Thenmozhi, B. R. Chakravarthi, A. Hegde, H. L. Shashirekha, R. Natarajan, S. Thavareesan, R. Sakuntharaj, K. Kalyanasundaram, C. Rajkumar, P. Shetty, and H. S. Kumar, “Overview of the Shared Task on Sentiment Analysis in Tamil and Tulu,” in *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, B. R. Chakravarthi, R. Priyadharshini, A. K. Madasamy, S.

- Thavareesan, E. Sherly, S. Rajiakodi, B. Palani, M. Subramanian, S. Cn, and D. Chinnappa, Eds. Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico: Association for Computational Linguistics, May 2025, pp. 732–738.
- [15] M. Prabhu, P. Shetty, and A. Nayak, “Sentiment Analysis in Tulu Using Ensemble Machine Learning Models,” in Proceedings of the 3rd Workshop on Speech and Language Technologies for Dravidian Languages (DravidianLangTech). Association for Computational Linguistics, 2023, pp. 67–74.
- [16] R. Kunderagi and R. Joshi, “Cross-Lingual Transfer for Sentiment Analysis in Low-Resource Dravidian Languages: A Case Study on Tulu,” in Proceedings of the 3rd Workshop on Speech and Language Technologies for Dravidian Languages (DravidianLangTech). Association for Computational Linguistics, 2023, pp. 75–83.
- [17] F. Koto, T. Beck, Z. Talat, I. Gurevych, and T. Baldwin, “Zero-shot sentiment analysis in low-resource languages using a multilingual sentiment lexicon,” in Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL), 2024.
- [18] J. Sm’id, P. P’rib’a’n, and P. Kr’al, “Improving Generative Cross-lingual Aspect-based Sentiment Analysis with Constrained Decoding,” 2025, arXiv preprint.
- [19] S. Manafi and N. Krishnaswamy, “Cross-lingual Transfer Robustness to Lower-resource Languages on Adversarial Datasets,” in LREC-COLING 2024, 2024.
- [20] Y. Wang, Z. Wang, N. Han, W. Wang, Q. Chen, H. Zhang, Y. Pan, and A. Nguyen, “Knowledge Distillation from Monolingual to Multi-lingual Models for Intelligent and Interpretable Multilingual Emotion Detection,” in WASSA 2024, 2024.
- [21] E. M. Ponti et al., “Survey of Cross-lingual Transfer Methods,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3204–3215.
- [22] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzm’an, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised Cross-lingual Representation Learning at Scale,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.
- [23] M. Artetxe and H. Schwenk, “Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 597–610, 2019.
- [24] G. Lample and A. Conneau, “Cross-lingual Language Model Pretraining,” in Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc., 2019, pp. 7057–7067.
- [25] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, “Xtreme: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization,” in ICML, 2020.
- [26] S. Ruder, E. Chau, J. Phang, J. Hu, G. Glava’s, E. M. Ponti, I. Vuli’c, R. Reichart, A. Søgaard, V. Srikumar et al., “Xtreme-r: Towards More Challenging and Realistic Cross-lingual Transfer Evaluations,” arXiv preprint arXiv:2104.07412, 2021.
- [27] “CodaLab Competition: Sentiment Analysis in Tamil and Tulu – dravidianlangtech@naacl2025”, <https://codalab.lisn.upsaclay.fr/competitions/20893>.
- [28] D. Bhatnagar, “Multilingual Sentiment Analysis Dataset for Indian Languages,” <https://huggingface.co/datasets/dhruv0808/indic-sentiment-analyzer>, 2024.
- [29] Durairaj Thenmozhi, Rathnakar Shetty P, Parameshwar R. Hegde, Anusha M D, Raksha Adyanthaya, Mohammed Fadhel Aljunid, Prasanna Kumar Kumaresan, and Bharathi Raja Chakravarthi. 2026. Findings of the Shared Task on Hope Speech Detection in Tulu. In Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. Association for Computational Linguistics.