GraMLID: GRU-Assisted Multilingual BERT for Word-Level Language Identification in Low-Resource Dravidian Texts

Krishna Tewari[1,*], Supriya Chanda[2] and Suhani Verma[3]

[1]Indian Institute of Technology (BHU), Varanasi, INDIA
[2]Bennett University, Greater Noida, INDIA
[3]Banasthali Vidyapith, Rajasthan, INDIA

Abstract

Word-level Language Identification in code-mixed social media text is a challenging task due to transliteration, script similarity, class imbalance, and noisy user-generated content. To address these challenges, we participated in the FIRE 2025 shared task on LID for five low-resource Dravidian languages (Kannada, Malayalam, Tamil, Telugu, and Tulu; alongside English). We propose a hybrid mBERT+GRU model that leverages multilingual transformer representations with recurrent sequence modeling. The model was trained with a learning rate of 2e-5, weight decay of 0.01, batch size of 16, and 150 epochs, with early stopping criteria to prevent overfitting. To handle class imbalance, we employed Focal Loss and oversampling strategies, while prediction cleaning was applied to remove irrelevant tags to ensure more accurate sequence labeling. Evaluation on the official shared task dataset, released by the organizers, demonstrates competitive performance across all languages. Our approach achieved peak accuracy of 0.94 for Kannada, with results of 0.89 for Tamil, 0.86 for Telugu, 0.85 for Tulu, and 0.83 for Malayalam. These findings highlight the effectiveness of combining transformer embeddings with lightweight recurrent layers, complemented by loss reweighting, prediction refinement, and early stopping, for robust LID in low-resource and code-mixed settings.

Keywords

Language Identification, Dravidian Languages, Code-Mixing, GRU, mBERT, Social Media

1. Introduction

The rapid proliferation of multilingual and code-mixed content on social media has amplified the need for robust *Language Identification (LID)* systems. At the word level, LID assigns a language label $y_t \in \mathcal{L}$ to each token $x_t \in \mathcal{X}$ in a sentence $X = (x_1, x_2, \ldots, x_T)$, where $\mathcal{L}$ denotes the set of languages under consideration. Such word-level annotation forms the foundation for downstream applications such as multilingual dialogue systems, sentiment analysis, and content moderation [1, 2].

The Indian linguistic landscape poses distinctive challenges for word-level LID [3]. Dravidian languages (Kannada, Malayalam, Tamil, Telugu, and Tulu) frequently appear in code-mixed text with English, often written in Roman transliteration. These languages share orthographic and phonological similarities, further complicating disambiguation. Compounding this difficulty, datasets from social media are inherently imbalanced, with high-resource languages such as English dominating, while under-represented languages like Tulu appear sparsely [4, 5]. These factors necessitate methods that can robustly handle noise, imbalance, and structural similarity in low-resource, multilingual contexts.

The Forum for Information Retrieval Evaluation (FIRE) has catalyzed progress in this domain through successive shared tasks on code-mixed LID. The CoLI-Tunglish 2023 task focused on Tulu-English word-level identification [6, 7], followed by CoLI-Dravidian 2024, which expanded the scope to multiple Dravidian languages mixed with English [8]. These initiatives laid the foundation for the FIRE 2025

Forum for Information Retrieval Evaluation, December 17-20, 2025, India

[*]Corresponding author.

krishnatewari.rs.cse24@itbhu.ac.in (K. Tewari); supriya.chanda@bennett.edu.in (S. Chanda);
btbte23017_suhani@banasthali.in (S. Verma)
0009-0005-6599-9956 (K. Tewari); 0000-0002-6344-8772 (S. Chanda)

shared task, which provides a benchmark dataset covering five Dravidian languages alongside English for word-level LID.

In this work, we present our participation in the FIRE 2025 shared task [9]. We propose a hybrid combining mBERT[1] and GRU[2] architecture that combines transformer-based contextual embeddings with lightweight recurrent sequence modeling. To further enhance robustness, we incorporate strategies for handling class imbalance and apply prediction refinement to ensure consistent labeling across code-mixed sequences.

The rest of the paper is structured as follows: Section 2 discusses related work; Section 3 describes the dataset; Section 4 presents the proposed methodology; Section 5 reports results and analysis; and Section 6 concludes with key findings.

2. Related Work

Over the decades, LID has progressed from rule-based statistical systems to modern neural and transformer models, particularly to handle code-mixed and low-resource language scenarios.

Initial LID systems predominantly used rule-based techniques, such as character-level n-gram models, which were quite effective for monolingual environments [10]. However, these methods struggled with code-mixed or transliterated text common in social media. Statistical models like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) offered improvements for high-resource languages [11], but their performance degraded significantly in noisy, short, social media style code-mixed scenarios.

Neural methods marked a step change. LSTM-based sub-word LID models for Indian languages and achieved robust performance on short sequences [12]. Bidirectional LSTMs (BiLSTM) for Hindi-English code-mixed texts yield notable improvements in handling noise and brevity in social media content [13].

The transformer era brought significant momentum to multilingual LID. Multilingual BERT (mBERT) [14] supports 104 languages, while XLM [15] improved cross-lingual learning for over 100 languages. India-focused variants like IndicBERT [16] and MuRIL [17] are tailored for Indian linguistic phenomena such as code-mixing and transliteration, improving performance in low-resource settings.

For structured sequence prediction, the BiLSTM-CRF architecture [18] has been widely deployed across sequence labeling tasks, establishing a precedent for combining contextual encodings with structure-aware decoding models. Despite these advancements, research focused specifically on low-resource Dravidian languages remains limited. The CoLI-Kanglish shared task (ICON 2022) provided a benchmark for Kannada-English word-level LID. BERT-based models achieve an 86% weighted F_1-score [19], while an overview report noted the highest macro F_1 around 0.62 [20]. Earlier work applied traditional classifiers such as KNN and SVM, reaching F_1-scores of around 0.58 [21]. Efforts further expanding to multiple Dravidian languages include Kannada-English dataset and benchmarked ML, DL, and transfer learning models, showing CoLI-ngrams achieved a macro F_1 of 0.64 [22].

Recently explored prompt engineering using GPT-3.5 Turbo for word-level LID in Dravidian languages, noting higher accuracy for Kannada over Tamil; further demonstrating the potential of large language model-based prompting for low-resource code-mixed LID [23]. Research in very low-resource and code-mixed LID often hinges on clever use of minimal data. Mandal and Sanand [24] proposed three strategies for code-mixed LID using minimal resources, achieving ensemble accuracy of approximately 92.6%.

In summary, while rule-based, neural, and transformer approaches have advanced LID significantly, their adaptation to code-mixed and low-resource Dravidian scenarios remains incomplete. Datasets for Tulu in particular are sparse, and class imbalance continues to degrade system performance. Few studies have combined hybrid architectures, imbalance-aware learning, and sequence refinement.

Our work addresses these gaps by introducing a hybrid mBERT+GRU model, incorporating Focal Loss, oversampling, and prediction cleaning for robust word-level LID across five low-resource Dravidian

[1] bert-base-multilingual-cased

[2] Gated Recurrent Unit

languages. In doing so, we build on the prior strengths while advancing resilience in challenging multilingual contexts.

3. Dataset

The dataset used in this study is released by the organizers of the FIRE 2025 Word-Level LID shared task. It consists of token-level annotated social media text across five low-resource Dravidian languages; Kannada, Malayalam, Tamil, Telugu, and Tulu, in addition to English.

Each language dataset varies in size and number of tag types, capturing a wide range of linguistic phenomena. For example, the Kannada dataset includes tags such as `kn`, `en`, `name`, and `loc`, while the Malayalam dataset also introduces additional tags like `num` and `plc`. Similarly, the Tamil dataset contains unique composite tags like `tmen` to capture mixed Tamil-English tokens, while the Tulu dataset contains cross-lingual overlaps with Kannada tokens.

Table 1 summarizes the number of training and validation sentences along with the tag types defined for each language in the FIRE 2025 LID dataset. Table 2 provides a detailed breakdown of tag frequency distributions across these splits, highlighting strong class imbalances across languages; for example, English tokens dominate in Kannada and Tulu, whereas native tokens are more prevalent in Tamil and Malayalam. Such disparities emphasize the necessity of strategies like loss reweighting and oversampling in our modeling pipeline. Finally, Table 3 presents representative example sentences from different Dravidian languages in the dataset, showcasing the complexity of multilingual, code-mixed text and further motivating the development of robust and adaptable models.

Table 1
Dataset Statistics: Number of Sentences and Tag Types in the FIRE 2025 LID Shared Task

Language	Train Sentences	Val Sentences	Tag Types
Kannada	147	115	6
Malayalam	126	93	7
Telugu	300	61	7
Tamil	300	61	6
Tulu	300	98	7

4. Methodology

In this section, we describe in detail the methodology followed in our work on word-level LID for Dravidian code-mixed texts. The pipeline is designed to handle the complex linguistic nature of code-switching, transliteration, and multilingual social media data. It consists of three main components: (i) preprocessing of raw data, (ii) model architecture combining mBERT and GRU, and (iii) training setup and optimization strategies. A stepwise overview of the architecture is summarized in Algorithm 1.

4.1. Preprocessing

Our preprocessing pipeline begin with the removal of unwanted characters, such as punctuation marks, special symbols, hashtags, and user mentions. While these features often serve as pragmatic markers in social media conversations, they do not directly contribute to LID at the token level. URLs are also stripped, as they are language-agnostic and introduce unnecessary noise into the embeddings.

Emojis, which are pervasive in online communication are removed. Unlike many NLP tasks where numbers can be discarded, in our case numeric tokens are retained because the dataset explicitly contained tags such as `num`, marking them as meaningful entities. This decision is essential to ensure consistency between the preprocessing pipeline and the annotation scheme.

Table 2
Dataset Statistics: Tag Frequency Distribution Across Training and Validation Splits

Language	Tag	Training	Development
Kannada	en	17,905	922
	kn	4,058	546
	name	1,353	50
	mixed	1,031	176
	loc	142	2
	other	2,588	47
Malayalam	ml	11,749	860
	en	5,762	407
	name	1,985	149
	mixed	761	61
	other	2,084	263
	num	601	43
	plc	123	8
Telugu	en	3,326	301
	te	1,248	57
	name	189	11
	mixed	64	32
	other	377	52
	num	260	5
	plc	27	–
Tamil	en	2,756	496
	tm	7,062	1,000
	name	1,134	160
	tmen	1,253	144
	other	82	1
	loc	11	–
Tulu	tu	11,644	1,251
	en	7,451	742
	kn	2,940	273
	name	1,511	135
	mixed	540	57
	loc	526	41
	other	658	85

Table 3
Sample Sentences and Token-Level Annotations for FIRE 2025 LID Dataset

Text	Token Labels
Namaskara friends nanu fine iddini	kn, en, kn, en, kn
Njan movie kandallo	mal, en, mal
Oru super film da	mal, en, en, tm
Chala bagundi performance	te, en
Tulu nataka performance super	tulu, tulu, en, en

Finally, redundant whitespace is normalized, ensuring uniform tokenization across sentences. The preprocessed data therefore represented a corpus that preserved meaningful linguistic and semantic markers while filtering noise irrelevant to the identification of language tags.

4.2. Model Architecture

The cornerstone of our approach is a hybrid architecture that combines the strengths of transformer-based encoders with recurrent sequence learners. Specifically, we employ the mBERT model as the base encoder and a GRU layer for sequential modeling.

4.2.1. Multilingual BERT Encoder

mBERT (`bert-base-multilingual-cased`) is a transformer-based model pre-trained on 104 languages using masked language modeling and next sentence prediction objectives. Its contextualized embeddings capture both inter-lingual and intra-lingual nuances, making it particularly suitable for multilingual and code-mixed scenarios. Each tokenized input sentence $X = (x_1, x_2, \dots, x_T)$ is passed through the mBERT encoder, producing contextual embeddings $E = (e_1, e_2, \dots, e_T)$, where each e_t captures bidirectional context around token x_t.

4.2.2. GRU Sequence Learner

Although transformers excel at capturing global context, they often underperform in modeling fine-grained sequential dependencies over long sequences, especially in noisy and code-mixed settings. To complement this, we integrate a GRU layer on top of mBERT embeddings. The GRU is a lightweight recurrent neural network variant that efficiently models temporal dependencies through its gating mechanisms. The GRU processes the embedding sequence E, producing hidden states $H = (h_1, h_2, \dots, h_T)$ that captures sequential context in a manner complementary to the transformer's global attention.

4.2.3. Classification Layer

The final hidden states H are passed through a fully connected layer, followed by a softmax function to produce probability distributions over the language tags for each token. Formally,

$$\hat{y}_t = \mathrm{softmax}(W \cdot h_t + b),$$

where W and b are trainable parameters of the classification layer. This ensures token-level predictions that aligned with the shared task's requirements.

4.2.4. Handling Data Imbalance and Cleaning Predictions

Code-mixed datasets suffer from high class imbalance, with English and dominant native languages heavily outnumbering minority tags such as named entities, numerals, or rare transliterated words. To mitigate this, we use Focal Loss, which dynamically down-weights easy-to-classify samples and places greater emphasis on harder, minority-class tokens. Additionally, oversampling of minority classes is performed during training to artificially balance the dataset.

Finally, a post-processing step called prediction cleaning is applied. This involved filtering out irrelevant labels such as `O` (outside any language span) or `sym` (symbols), which occasionally appear in predictions despite not being semantically meaningful for the downstream evaluation.

The complete stepwise procedure of the model pipeline is presented in Algorithm 1.

Algorithm 1 Proposed mBERT+GRU Framework for Word-Level LID

```
Input: Tokenized code-mixed sentence $X = (x_1, x_2, \dots, x_T)$
Preprocessing: Clean text (remove URLs, hashtags, punctuation, mentions, emojis; retain numbers)
Obtain contextualized embeddings $E = \mathrm{mBERT}(X)$
Pass embeddings through GRU layer: $H = \mathrm{GRU}(E)$
Apply fully connected + softmax: $\hat{y}_t = \mathrm{softmax}(W \cdot h_t + b)$
Compute loss using Focal Loss with dynamic class weighting
Oversample minority classes during training
Perform prediction cleaning to remove irrelevant tags (O, sym)
Output: Predicted sequence labels $\hat{Y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T)$
```

4.3. Training Setup

The model is trained end-to-end with token-level supervision from the FIRE 2025 LID shared task dataset. To optimize performance, we employ several training strategies, which we describe below.

We use the Adam optimizer with decoupled weight decay (AdamW), which has become the de-facto standard for transformer-based fine-tuning. The learning rate is initialized at 2×10^{-5}, a value empirically tuned for stability, and weight decay is set at 0.01 to prevent overfitting. A batch size of 16 is adopted, balancing computational efficiency with gradient stability.

The model is trained for a maximum of 150 epochs. However, to mitigate overfitting and reduce unnecessary computation, we employ an early stopping criterion. Training is terminated once the validation loss plateaued for 3 consecutive epochs, ensuring that the model retains generalizable performance without memorizing training data.

Given the nature of social media text, which can range from short phrases to longer posts, we set the maximum sequence length to 512 tokens. This value ensures coverage for most sentences without truncation. The WordPiece tokenizer associated with mBERT is used to handle out-of-vocabulary tokens, ensuring robust subword segmentation across languages.

The choice of Focal Loss is crucial in addressing dataset imbalance. Unlike traditional cross-entropy, which treats all tokens equally, Focal Loss modulates the contribution of easy versus hard samples, with a focusing parameter γ that down-weights well-classified examples. This ensures that rare labels such as numerals or location names are not overshadowed by dominant classes. Oversampling further complemented this by artificially replicating underrepresented class instances during training, balancing the gradient contributions across labels.

5. Results

We evaluate the performance of our proposed mBERT+GRU framework on the FIRE 2025 LID shared task datasets. The experiments are carried out on the validation datasets provided by the organizers, where we computed detailed classification reports (per-class Precision, Recall, F_1-Score, Accuracy, and Support). These are reported for each of the five Dravidian languages separately. The final test set results are obtained from the official leaderboard and are summarized at the end of this section.

We observe that performance is consistently strong for high-frequency classes such as `ENGLISH` and the major Dravidian language tag in each dataset, whereas minority categories (e.g., `Location`, `Number`, `Other`, `Place`) tend to have lower F_1-scores due to data imbalance.

On the Kannada validation set (Table 4), the model achieves an overall accuracy of 0.9079. It demonstrates strong recognition of English (F_1 = 0.97) and Kannada (F_1 = 0.89), although categories such as "other" and "name" are comparatively weaker. For Tulu (Table 5), the accuracy was 0.8177, with high scores for English (F_1 = 0.90) and Tulu (F_1 = 0.87), while mixed-language tokens remain particularly challenging (F_1 = 0.48).

In the case of Telugu (Table 6), the model obtains an overall accuracy of 0.7948. Performance is excellent for English (F_1 = 0.90) and mixed tokens (F_1 = 0.98), but categories with sparse representation such as "number" (F_1 = 0.33) and "other" (F_1 = 0.42) prove difficult to classify reliably. Similarly, the Tamil dataset (Table 7) yields an accuracy of 0.8989, where Tamil (F_1 = 0.93) and English (F_1 = 0.93) are predicted with high consistency, whereas less frequent categories like "location" (F_1 = 0.65) show reduced performance.

Finally, the Malayalam dataset (Table 8) reached an overall accuracy of 0.8705. The model performs particularly well on Malayalam (F_1 = 0.94) and English (F_1 = 0.90), but struggles with underrepresented categories such as "place" (F_1 = 0.00) and "mixed" tokens (F_1 = 0.35). Taken together, these results highlight the robustness of the approach in handling high-resource categories, while underscoring persistent challenges in dealing with rare or highly imbalanced classes.

Table 4
Classification Report: Kannada (Validation Data)

Class	Precision	Recall	F_1-Score	Accuracy	Support
en	0.9596	0.9800	0.9697	0.9079	500
kn	0.8896	0.8937	0.8916	0.9079	250
other	0.6842	0.5909	0.6333	0.9079	44
name	0.7636	0.8409	0.8000	0.9079	88
mixed	0.8070	0.7011	0.7500	0.9079	87
location	0.7273	0.6154	0.6667	0.9079	13

Table 5
Classification Report: Tulu (Validation Data)

Class	Precision	Recall	F_1-Score	Accuracy	Support
English	0.9057	0.8902	0.8979	0.8177	265
Tulu	0.8844	0.8552	0.8695	0.8177	154
Other	0.5806	0.5625	0.5714	0.8177	16
Mixed	0.4615	0.5000	0.4809	0.8177	10

Table 6
Classification Report: Telugu (Validation Data)

Class	Precision	Recall	F_1-Score	Accuracy	Support
ENGLISH	0.9208	0.8750	0.8970	0.7948	288
TELUGU	0.7288	0.8600	0.7891	0.7948	200
MIXED	0.9692	0.9840	0.9765	0.7948	250
NUMBER	0.2857	0.4286	0.3333	0.7948	14
OTHER	0.4444	0.4074	0.4225	0.7948	27
NAME	0.6875	0.5789	0.6286	0.7948	19

Table 7
Classification Report: Tamil (Validation Data)

Class	Precision	Recall	F_1-Score	Accuracy	Support
en	0.9212	0.9432	0.9321	0.8989	370
tm	0.9209	0.9380	0.9293	0.8989	350
mixed	0.9256	0.8659	0.8947	0.8989	82
location	0.6579	0.6364	0.6465	0.8989	22
other	0.7222	0.5909	0.6500	0.8989	22
name	0.8571	0.8571	0.8571	0.8989	14
place	0.7143	0.6250	0.6667	0.8989	8

5.1. Leaderboard Results

The final system submissions were evaluated on the official test sets, and the scores were reported on the shared task leaderboard. The results across the five languages are summarized in Table 9. Among the languages, Kannada achieved the highest score of 0.94, followed by Tamil with 0.89, and Telugu with 0.86. Tulu and Malayalam obtained scores of 0.85 and 0.83, respectively. These leaderboard outcomes are consistent with the validation results, reflecting strong performance in high-resource languages such as Kannada and Tamil, while relatively lower but competitive results were observed in Malayalam and Tulu.

5.2. Error Analysis

Despite strong overall performance, the model shows weaknesses in handling minority categories such as place, number, and other, where limited training instances and class imbalance reduce reliability. Malayalam and Tulu exhibit comparatively lower scores, largely due to sparse data and script overlap

Table 8
Classification Report: Malayalam (Validation Data)

Class	Precision	Recall	F_1-Score	Accuracy	Support
ENGLISH	0.9282	0.9274	0.9278	0.8511	292
MALAYALAM	0.9024	0.9066	0.9045	0.8511	242
MIXED	0.9048	0.8444	0.8736	0.8511	90
NAME	0.7500	0.6000	0.6667	0.8511	10
OTHER	0.6250	0.7143	0.6667	0.8511	7
NUMBER	0.6667	0.5000	0.5714	0.8511	6

leading to higher confusion among closely related tokens. The GRU layer, while effective for short dependencies, struggles with long or abrupt language switches typical of social media text. Moreover, mBERT's general-domain pretraining limits its ability to fully capture domain-specific transliterations and informal expressions, suggesting that domain-adaptive fine-tuning and richer cross-lingual representations could further enhance robustness.

Table 9
Leaderboard results on the official test sets across languages.

Language	**File Name**	**Score**
Kannada	Test_IReL_Kan_Run1.zip	0.94
Malayalam	Test_IReL_mal_Run1.zip	0.83
Telugu	Test_IReL_tl_Run1.zip	0.86
Tamil	Test_IReL_Tm_Run1.zip	0.89
Tulu	Test_IReL_Tulu_Run1.zip	0.85

6. Conclusion

In this paper, we presented a hybrid mBERT+GRU model for word-level LID in code-mixed Dravidian social media text, addressing challenges of transliteration, noisy input, and class imbalance through focal loss, oversampling, and prediction refinement. Our system achieved strong leaderboard results, peaking at 0.94 accuracy for Kannada, alongside competitive scores for Tamil (0.89), Telugu (0.86), Tulu (0.85), and Malayalam (0.83), demonstrating the effectiveness of combining multilingual transformer embeddings with lightweight sequential modeling. While the approach proved robust across languages, relatively lower performance in Malayalam and Tulu highlights the limitations posed by data scarcity and script overlap. Future work should explore cross-lingual pretraining with domain-specific corpora, advanced sequence encoders such as graph or attention-based architectures, and transfer learning across related Dravidian languages to enhance generalization. Further emphasis should also be placed on model-agnostic post-processing and deployment-oriented strategies for reliable, real-time LID in multilingual user-generated content.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Lui, T. Baldwin, Automatic identification of multilingual documents, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 658–667.

[2] S. Chanda, K. Tewari, A. Mukherjee, S. Pal, Leveraging chatgpt and xlm-roberta for sarcasm detection in dravidian code-mixed languages, in: Proceedings of FIRE (Working Notes), Forum for Information Retrieval Evaluation, 2024, India, 2024. URL: https://ceur-ws.org/Vol-4054/T4-14.pdf.

[3] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, H. L. Shashirekha, G. Sidorov, A. Gelbukh, Overview of coli-kanglish: Word level language identification in code-mixed kannada-english texts at icon 2022, in: Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, 2022, pp. 38–45.

[4] R. Prathiba, R. Kannan, Language identification in code-mixed data: Challenges and approaches, Journal of Intelligent Systems (2020).

[5] A. Hegde, M. D. Anusha, S. Coelho, H. L. Shashirekha, 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 (SIGUL), European Language Resources Association (ELRA), Marseille, France, 2022, pp. 33–40.

[6] A. Hegde, F. Balouchzahi, S. Coelho, S. H L, H. A. Nayel, S. Butt, Coli@fire2023: Findings of word-level language identification in code-mixed tulu text, in: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE '23, Association for Computing Machinery, New York, NY, USA, 2024, p. 25–26. URL: https://doi.org/10.1145/3632754.3633075. doi:10.1145/3632754.3633075.

[7] A. Hegde, F. Balouchzahi, S. Coelho, H. L. Shashirekha, H. A. Nayel, S. Butt, Overview of coli-tunglish: Word-level language identification in code-mixed tulu text at fire 2023, in: Forum for Information Retrieval Evaluation (FIRE 2023) Working Notes, 2023, pp. 179–190.

[8] A. Hegde, F. Balouchzahi, S. Butt, S. Coelho, K. G, H. S. Kumar, S. D, S. H. L., A. Agrawal, Coli@fire2024: Findings of word-level code-mixed language identification in dravidian languages, in: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE '24, Association for Computing Machinery, New York, NY, USA, 2025, p. 7–10. URL: https://doi.org/10.1145/3734947.3735663. doi:10.1145/3734947.3735663.

[9] A. Hegde, F. Balouchzahi, S. Butt, S. Coelho, S. Hosahalli Lakshmaiah, A. Agrawal, Overview of CoLI-Dravidian 2025: Word-level Code-Mixed Language Identification in Dravidian Languages, in: Forum for Information Retrieval Evaluation FIRE - 2025, 2025.

[10] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, A statistical approach to language identification, Computational Linguistics 18 (1992) 611–620.

[11] B. Hughes, T. Baldwin, M. Lui, Re-examining language identification, Journal of Computational Linguistics 32 (2006) 45–60.

[12] A. Joshi, S. Negi, N. Goel, L. Singh, M. Shrivastava, Towards sub-word level language identification for indian languages, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016, pp. 1–10.

[13] Y. Zhang, Z. Yang, J. Qi, Deep learning for code-mixed language identification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018, pp. 2246–2255.

[14] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[15] A. Conneau, U. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, A. Joulin, M. Koepke, Cross-lingual language model pretraining, in: Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 7057–7067.

[16] D. Kakwani, A. Kunchukuttan, S. Gella, P. Bhattacharyya, M. Gokhale, A. Agarwal, R. Bhat, N. Kedia, A. Sharma, M. Kumar, IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages, in: Proceedings of the 12th Language Resources and Evaluation Conference (LREC), 2020, pp. 1490–1499.

[17] S. Khanuja, A. Kunchukuttan, S. Kumar, M. Singh, S. Prasad, S. Gella, P. Bhattacharyya, A. Kumar, MuRIL: Multilingual representations for indian languages, arXiv preprint arXiv:2103.10730 (2021).

[18] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv preprint

arXiv:1508.01991 (2015).

[19] P. Deka, N. J. Kalita, S. K. Sarma, Bert-based language identification in code-mix kannada-english text at the coli-kanglish shared task, in: ICON 2022 Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, ACL, 2022, pp. 12–17.

[20] F. Balouchzahi, S. Butt, A. Hegde, et al., Overview of coli-kanglish: Word level language identification in code-mixed kannada-english texts at icon 2022, in: ICON 2022 Shared Task on Word Level Language Identification, ACL, 2022, pp. 38–45.

[21] M. Shahiki Tash, Z. Ahani, A. Tonja, et al., Word level language identification in code-mixed kannada-english texts using traditional machine learning algorithms, in: ICON 2022 Shared Task on Word Level Language Identification, ACL, 2022, pp. 25–28.

[22] H. Shashirekha, F. Balouchzahi, M. Anusha, et al., Coli-machine learning approaches for code-mixed language identification at the word level in kannada-english texts, in: CoLI shared task workshop, 2022.

[23] A. Deroy, S. Maity, Prompt engineering using gpt for word-level code-mixed language identification in low-resource dravidian languages, arXiv preprint arXiv:2411.04025 (2024).

[24] S. Mandal, S. Sanand, Strategies for language identification in code-mixed low resource languages, arXiv preprint arXiv:1810.07156 (2018).