Exploring Language-Specific Characteristics for Word-level Language Identification in Dravidian Languages

Rachana Nagaraju[*,†], H L Shashirekha[†]

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

Language Identification (LI) is essential for many Natural Language Processing (NLP) applications, including sentiment analysis, machine translation, and information retrieval. Reliable LI is crucial in multilingual and informal communication settings, where language boundaries can blur. This is particularly true in India, where social media users often create code-mixed text combining English with regional and/or local languages. The Dravidian languages — Kannada, Tamil, Malayalam, Telugu, and Tulu — have complex structures and are frequently Romanized and/or mixed with English language in digital conversations. Processing code-mixed text in these languages is challenging as they are low-resourced. To tackle these issues, the shared task on 'Word-level Language Identification for Code-Mixed Dravidian Languages' by CoLI-Dravidian @ Forum for Information Retrieval Evaluation (FIRE) 2025 provided word-level annotated datasets in Kannada, Tamil, Malayalam, Telugu, and Tulu, with the aim of fostering the development of strong LI systems. In this paper, we — team **MUCS** model the Word-level LI task as a classical sequence labeling problem and describe a Conditional Random Field (CRF)-based pipeline, blending various lexical and contextual features, to address the challenges of the shared task. Our system performed well across all languages, achieving Macro F1 scores of **0.9040** in Kannada (**6th** rank), **0.5955** in Tamil (**8th** rank), **0.7620** in Malayalam (**6th** rank), **0.7289** in Telugu (**4th** rank), and **0.7963** in Tulu (**5th** rank). These results show that a carefully designed classical sequence labeling approach can remain competitive with other methods, even in noisy and code-mixed multilingual settings.

Keywords

Language Identification, Code-Mixing, Dravidian Languages, Conditional Random Fields, Multilingual NLP

1. Introduction

LI involves identifying the language of a text fragment, ranging from a full document to just one word. This task is a crucial first step in many NLP applications like machine translation, sentiment analysis, and named entity recognition [1, 2]. Although LI at the sentence or document level is quite advanced, identifying the language at the word-level is still complicated, particularly in informal, multilingual contexts. This complexity is especially evident in Indian languages, where multilingualism is common. Digital conversations often include significant code-mixing, which is the blending of words and structures from different languages including English in the same sentence or utterance [3]. Code-mixed texts on social media are typically informal, inconsistent in spelling, and heavily transliterated. They are often written in Roman and/or native script, regardless of the user's native language. These traits greatly limit the effectiveness of traditional NLP tools, as they expect monolingual and well-formed input.

Dravidian languages are a major language family in South India and parts of Sri Lanka. Key members of this family include Kannada, Tamil, Telugu, Malayalam, and Tulu. These languages are rich in morphology and highly agglutinative, but they lack sufficient computational resources. Despite having millions of native speakers, they are labeled as low-resource languages in NLP due to lack of large annotated datasets, pre-trained models, and linguistic tools. Much of the online content in these

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

[*]Corresponding author.

[†]These authors contributed equally.

rachananagaraju20@gmail.com (R. Nagaraju); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

0000-0002-9421-8566 (H. L. Shashirekha)

Table 1
Representative Samples and Class-wise Distribution of Samples in Train Set

Language	Word	Tag	No. of Samples
Kannada	edanthu	KANNADA	4,058
	like	ENGLISH	17,905
	*	SYMBOL	3,833
	usemadu	MIXED	1,031
	kubi	OTHER	2,588
	dvg	LOCATION	142
	shiva	NAME	1,353
Malayalam	vesham	MALAYALAM	11,749
	police	ENGLISH	5,762
	Ettante	NAME	1,985
	Shorttum	MIXED	761
	irukku	OTHER	2,084
	7	NUMBER	601
	Kochi	PLACE	123
Telugu	chusthe	TELUGU	1,248
	blockbuster	ENGLISH	3,326
	nanis	NAME	189
	optioni	MIXED	64
	4	NUMBER	260
	*	SYM	789
	Vizag	PLACE	27
	vachesaru	OTHER	377
Tamil	Kudukura	TAMIL	7,062
	Features	ENGLISH	2,756
	dinesh	NAME	1,134
	logicke	MIXED	1,253
	*	SYMBOL	1,189
	Madurai	LOCATION	11
	pakkam	OTHER	82
Tulu	Kuku	TULU	11,644
	ajji	KANNADA	2,940
	super	ENGLISH	7,451
	Ko	OTHER	658
	.	SYM	4,254
	Mangalore	LOCATION	526
	superadh	MIXED	540
	venkat	NAME	1,511

languages, particularly on social media, is informal and often mixed with English, which creates additional challenges for automated processing.

To tackle the challenges of word-level LI posed by code-mixed text in Dravidian languages on social media, the CoLI-Dravidian @ FIRE 2025[1] shared task [4] introduced a Word-level LI challenge. This initiative aims to enable more accurate linguistic analysis in multilingual contexts. Participants are given word-level annotated datasets in five languages — Kannada, Tamil, Telugu, Malayalam, and Tulu — with each word tagged as belonging to a specific language (e.g., TAMIL, MALAYALAM) or a functional class (e.g., ENGLISH, NAME, SYM, MIXED, OTHER). The goal of this task is to evaluate and improve LI systems that perform well in noisy, low-resource, and morphologically complex situations. Table 1 presents the tags and tag-wise distribution of words in five languages of the Train set. It can be observed that the Train sets of all the languages are imbalanced.

We, team - MUCS, participated in this shared task using a traditional yet effective approach. We modeled word-level LI task as a classical sequence labeling problem and developed a pipeline that

[1]https://www.codabench.org/competitions/7902/

included thoughtfully designed lexical, orthographic, and contextual features, to train a CRF model for each language. Our code is available on GitHub[2] to reproduce the results and explore further. This setup allows for better adaptation to the morphology and word patterns of the individual languages. While neural models have recently gained popularity in NLP, we believe that language-specific feature-based CRF models remain effective, particularly in low-resource and noisy conditions and well-designed features can surpass end-to-end neural systems [5, 6]. In this paper, we describe our complete approach for word-level LI in Dravidian code-mixed data, covering feature extraction, model training, and evaluation. We also provide detailed analyses of our system's behavior across languages and explore its potential for real-world multilingual applications.

The subsequent sections of this paper details the related works (Section 2), methodology (Section 3), experiments, results, and implications of our approach (Section 4), declaration on generative AI (Section 5) followed by conclusion and future works (Section 6).

2. Related Works

LI in code-mixed and multilingual settings has received growing attention in recent years, particularly in low-resourced languages such as those in the Dravidian family. Several shared tasks have previously focused on word-level LI in code-mixed Indic texts. Notable among them are *CoLI-Kanglish: Word-Level Language Identification in Code-mixed Kannada-English Texts* at ICON 2022 [7, 8], *CoLI-Tunglish: Word-level Language Identification in Code-mixed Tulu Texts* at FIRE 2023 [9, 10], and *CoLI-Dravidian: Word-level Code-Mixed Language Identification in Dravidian Languages* at FIRE 2024 [11]. These tasks have laid the groundwork for advancing research in LI in multilingual and morphologically rich environments. Several researchers have explored traditional Machine Learning (ML), Deep Learning (DL), and transformer-based models for word-level and sentence-level LI tasks. Few notable works are described below:

Chakravarthi et al. [12] introduced *DravidianCodeMix* dataset, which contains social media comments in Tamil-English, Kannada-English, and Malayalam-English, for sentiment analysis and offensive language detection. They reported baseline experiments using classical ML models and DL architectures, allowing for comparative benchmarks in later studies. Shimi et al. [13] conducted a comparative study of different ML algorithms (Naive Bayes, Support Vector Machines, Logistic Regression, and Random Forest), alongside transformer models such as BERT and mBERT. Their study focused on Tamil and Malayalam and found that while classical models achieved accuracy between 85-89%, transformer models reached up to 98% accuracy. This highlighted the strength of pre-trained language models on both monolingual and code-mixed data.

The *VarDial Dravidian Language Identification* shared task [14] evaluated various methods for identifying languages in code-mixed data. The organizers of the shared task compared character n-gram models with contextual transformers like RoBERTa. While transformers are popular, character-based models showed strong macro F1 scores, particularly in low-resource settings involving Kannada, Tamil, and Malayalam. Deroy and Maity [15] explored prompt-based learning with GPT-3.5 Turbo for word-level LI in code-mixed Kannada and Tamil. The model showed high precision for English and Kannada words but faced challenges with mixed-language segments. This indicated the limitations of prompt-only approaches in complex linguistic environments like intra-word code-mixing. Hande et al. [16] used transfer learning models such as ULMFiT and BERT for detecting offensive language in code-mixed Tamil, Malayalam, and Kannada. While the main focus was not on LI, their pipeline involved language-aware preprocessing and word-level modeling strategies.

Mandalam and Sharma [17] trained Logistic Regression and LSTM networks with Term Frequency-Inverse Document Frequency (TF-IDF) for sentiment analysis on code-mixed Tamil and Malayalam texts in FIRE 2020. Their results showed that neural models performed better when they used domain-specific pre-processing. Their setup included modules for identifying intermediate languages to help with classification. Saumya et al. [18] experimented with lightweight models like Naive Bayes and

[2] https://github.com/rachanabn20/CoLI-Dravidian-FIRE-2025

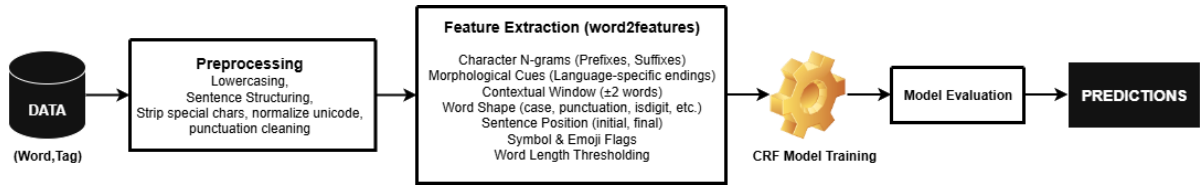


Figure 1: Architecture of the CRF-based Word-level Language Identification Pipeline

shallow neural networks, using n-gram features for detecting offensive content in Tamil-English and Malayalam-English datasets. Their study shows that simple lexical models worked well in noisy social media environments. The IndicNLP@KGP team [19] participated in the DravidianLangTech-EACL 2021 shared task using a combination of AWD-LSTM and transformer models for word-level classification. Their efforts led to F1 scores of 0.97 for Malayalam and 0.77 for Tamil. This performance demonstrates their strength in handling morphologically rich, code-mixed inputs.

Informed by these prior works, our system adopts a feature-rich CRF framework tailored to handle intra-word code-mixing and symbol/other word categorization. Unlike many previous approaches focused on sentence-level labels or sentiment detection, our system is optimized for fine-grained word-level classification as demanded by the CoLI-Dravidian @ FIRE 2025 task.

3. Methodology

This section describes our proposed system pipeline for the CoLI-Dravidian @ FIRE 2025 shared task. We employed a CRF model to perform word-level LI in code-mixed Dravidian social media text. CRF model is well-suited for sequential data with strong inter-word dependencies and provides interpretable feature-based modeling. Figure 1 visualizes the architecture of our CRF-based pipeline for word-level LI. We transform the given (word, tag) pairs into sentence structure based on sentence boundaries. Each sentence is then processed as a sequence of words and each word is transformed into a feature vector using handcrafted features (e.g., character n-grams, word shape, contextual clues). These vectors are fed to the CRF layer, to predict the most likely sequence of language tags by modeling dependencies between neighboring tags. The steps involved in building CRF model are described below:

3.1. Language-Specific Characteristics

Tamil, Kannada, Malayalam, Telugu, and Tulu languages are morphologically rich and exhibit agglutinative word formation where words are formed by stringing two or more morphemes without altering them, making morphology highly regular. Code-mixed data involving these languages often blend native morphology with English words or named entities. For instance, suffix patterns in Tamil (e.g., *-kura*, *-vanga*, *-ungal*) or Telugu (e.g., *-chusthe*, *-andi*, *-nunchi*) provide cues for LI. Suffixes such as *-andi* in Telugu or *-vanga* in Tamil often indicate politeness, intent, or action and are highly language-specific. Kannada exhibits characteristic endings like *-alli* (locative), *-ige* (dative), and *-iddu* (copula), while Malayalam frequently uses markers such as *-ille* (negation), *-kal* (plural), and *-um* (conjunctive/also). Tulu, though less resourced, shows identifiable suffixes like *-d* (past tense), *-er* (plural), and *-ndu* (emphatic). In contrast, named entities and borrowed English terms may appear uniformly across languages, adding ambiguity for LI. These orthographic and morphological cues thus play a vital role in LI within code-mixed settings.

3.2. Feature Extraction

The core of our model pipeline is `word2features()` function, which generates handcrafted features for each word in a sentence. Inspired by prior CRF-based NER[3] and LID systems [5], the function

[3]Lample et al. (2016), introduced neural sequence labeling architectures for NER, combining word- and character-level features

incorporates both word-level and context-level properties. For a given word w_i in a sentence $S = \{w_1, w_2, ..., w_n\}$, we extract:

- **Current word features:**
  - Lowercased form of the word if the text is in Roman script.
  - Character prefixes/suffixes (1 to 3 characters) - short prefixes/suffixes help capture morphological or inflectional endings relevant to agglutinative languages.
  - Word shape - all-capitals, all-lowercase, title case - casing features help distinguish acronyms, named entities, and sentence boundaries.
  - Digit/emoji/URL/symbol flags - flags for emojis or symbols are useful in social media or informal text settings.
  - Length of the word.
  - Language-specific morphological clues (e.g., frequent suffixes like *-vanga* in Tamil, *-andi* in Telugu, *-an* or *-amma* in Malayalam, *-nu* or *-ra* in Kannada, and *-da* or *-du* in Tulu).
- **Contextual features:**
  - Previous and next word features - contextual windows (typically size 1 or 2) helps to capture the features better.
  - Position of the word in the sentence - sentence-initial or sentence-final positions can hint at part-of-speech or discourse-level roles.

The `word2features()` function constructs a rich feature vector for each word, which is then passed to the CRF model, enabling sparse yet interpretable modeling.

3.3. Model Training

We used `sklearn-crfsuite`[4] - a fast CRF implementation as a Python wrapper for labeling sequence data, to train CRF models for each language using language-specific features of that language and the corresponding tags. Words are classified into one of the tags based on their form and context. To ensure reproducibility and robustness across the linguistically diverse Dravidian languages, hyperparameters are selected through a combination of insights from prior CRF-based sequence labeling work [11] and systematic empirical tuning on held-out validation sets for each language. Specifically, we performed grid search over L1 ($c1$) and L2 ($c2$) regularization coefficients, testing values in the ranges 0.01–0.2 for $c1$ and 0.001–0.02 for $c2$, while monitoring F1 scores to balance generalization and overfitting. This per-language tuning revealed that the shared configuration ($c1 = 0.1$, $c2 = 0.01$) yielded the optimal trade-off across all datasets, with minimal variance (e.g., <2% F1 score fluctuation between languages). Other parameters, such as max iterations (200) and context window size (2), are similarly validated on validation splits to avoid underfitting on agglutinative patterns while preventing noise from larger windows. The selected hyperparameters are summarized in Table 2.

One of the practical advantages of using CRF models, particularly in our pipeline, is their training efficiency and low computational overhead. Using `sklearn-crfsuite` on Google Colab, CRF model for each language was trained within **5–7 minutes**, even when using feature-rich configurations. In contrast to large transformer-based models that often require GPU acceleration and hours of training time, our CRF-based approach is CPU-friendly, memory-efficient, and well-suited for low-resource environments. This makes our system accessible to researchers with limited computing infrastructure while still achieving competitive accuracy in code-mixed setting.

[4] https://sklearn-crfsuite.readthedocs.io/en/latest/

Table 2
Hyperparameters used in Training CRF Model

Parameter	Value
Algorithm	L-BFGS
c1 (L1 penalty)	0.1
c2 (L2 penalty)	0.01
Max iterations	200
All possible transitions	True
Feature frequency threshold	1
Context window size	2
Character N-gram range	1–3
Lowercasing	True

Table 3
Language-wise Statistics of the Datasets

Language	Train Set	Validation Set	Test Set
Kannada	30,910	2,016	2,075
Malayalam	25,995	2,008	1,997
Telugu	6,280	515	494
Tamil	13,514	1,984	2,066
Tulu	29,524	3,006	3,283

4. Experiments and Results

In this section, we describe the dataset, experimental setup, evaluation metrics, and the results of our word-level LI models. We performed experiments using the datasets provided by the organizers of CoLI-Dravidian @ FIRE 2025 shared task for five languages: Kannada, Malayalam, Telugu, Tamil, and Tulu. The task is framed as a classical sequence labeling problem, with sequence tags that include native language, English, named entities, symbols, mixed words, and other semantic categories. The dataset is pre-tokenized into words in Roman script and annotated with word-level tags. Table 3 summarizes the statistics of the datasets in terms of the number of words for the five languages across Train, Validation, and Test sets, for each language.

4.1. Results

The proposed CRF model is evaluated on the Test sets of each language based on Macro F1-score which is well-suited for imbalanced class distributions. Table 4 presents the Precision, Recall and Macro F1-scores obtained by our models for the five Dravidian languages for both Validation and Test sets. Our system performed consistently well, securing competitive ranks in each language track. Notably, we achieved 4^{th} rank in Telugu, 5^{th} rank in Tulu, 8^{th} in Tamil and 6^{th} rank in both Kannada and Malayalam tracks. Figure 2 shows performances of the models submitted by all participants of the shared task for each language. It is evident that classical sequence labeling approaches, when engineered with domain-relevant features, can compete effectively against neural and multilingual transformer-based systems. Overall, the results of our models reaffirm that even in the presence of multilinguality, informal orthography, and limited data, structured models like CRFs when paired with linguistic insights can deliver reliable and interpretable performance.

4.2. Error Analysis

Despite overall promising performance, our models exhibited several notable misclassifications in all five languages. To understand the behavior of our system across the languages, we present the confusion matrices in Figure 3. These matrices highlight common error patterns and dominant misclassification trends. Frequent confusions are observed between script-similar pairs (e.g., Kannada–Tulu), class-

Table 4
Performances of the Proposed CRF models on Validation and Test Sets

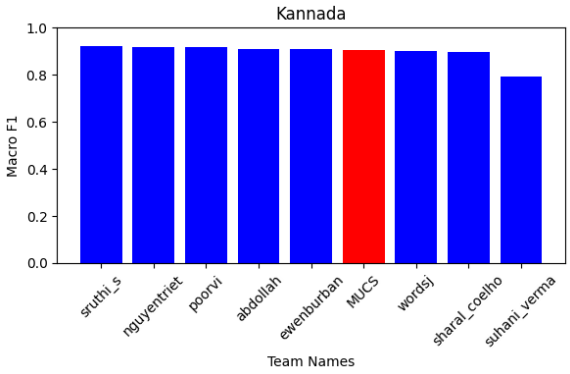
Language	Validation Set			Test Set			
	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Rank
Kannada	0.90	0.76	0.81	0.91	0.90	0.9040	6^{th}
Malayalam	0.85	0.79	0.81	0.84	0.72	0.7620	6^{th}
Telugu	0.85	0.79	0.81	0.74	0.72	0.7289	4^{th}
Tamil	0.73	0.71	0.72	0.63	0.61	0.5955	8^{th}
Tulu	0.85	0.75	0.79	0.86	0.74	0.7963	5^{th}

Table 5
Sample Misclassifications with Actual and Predicted Tags

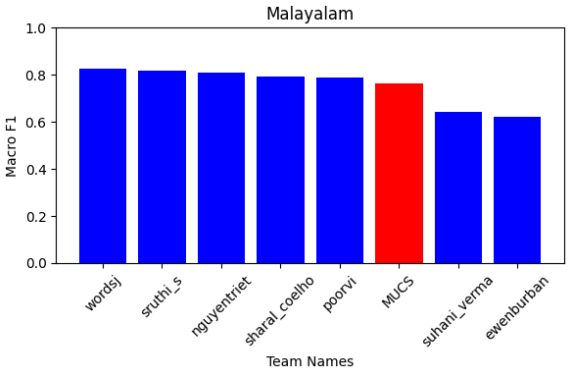
Language	Word	Actual Tag	Predicted Tag
Kannada	movieolle	MIXED	KANNADA
	babydu	MIXED	ENGLISH
	hudugidu	KANNADA	MIXED
	pak	OTHER	LOCATION
Malayalam	kazhinuu	NAME	MALAYALAM
	100m	NUMBER	ENGLISH
	Gandhi	NAME	OTHER
Telugu	thalli	TELUGU	ENGLISH
	sukanga	TELUGU	MIXED
	mi	TELUGU	OTHER
Tamil	kb	OTHER	TAMIL
	Sumara	TAMIL	NAME
	mai	OTHER	ENGLISH
Tulu	yearda	MIXED	TULU
	jasthi	TULU	KANNADA
	dithijiddu	TULU	MIXED
	mint	TULU	ENGLISH

sparse categories (e.g., MIXED), and semantically overlapping tags (e.g., ENGLISH and NAME). Table 5 lists manually inspected word-level prediction errors to highlight specific model confusions for each language as given below:

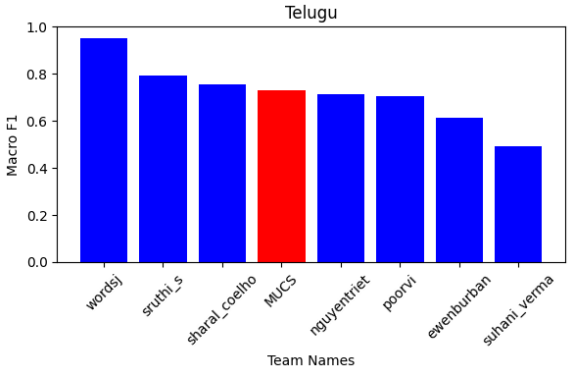
- **Kannada:** The model struggles to disambiguate mixed-language words that contain Kannada morphemes but exhibit lexical borrowing (e.g., MIXED WORD "babydu" misclassified as ENGLISH). CRF dependency on local context also causes OTHER or MIXED tokens to be absorbed into dominant language tags like KANNADA. Improved modeling of bilingual spans or hybrid morphemes could reduce these inconsistencies.
- **Malayalam:** Overlapping of native suffixes (*-an*, *-amma*) and common name endings leads to confusion between MALAYALAM and NAME tags. Using an external name lexicon could help disambiguate such tokens. Stylized English forms also confuse the system when they are morphologically similar to Malayalam (e.g., "kazhinuu").
- **Telugu:** Errors are dominated by transliterated or borrowed forms in Roman script. Models fail to separate phonetically similar tokens such as "thalli" (native) and "tally" (borrowed from English). Subword normalization and phonetic-aware features can mitigate this.
- **Tamil:** Confusion between TAMIL and MIXED tags occurs frequently in constructs like "call pannunga", where an English verb stem merges with a Tamil suffix. Misclassifications between NAME and ENGLISH tags occur in named entities using Latin script — e.g., "Sumara" or "Thomas".
- **Tulu:** Due to script similarity with Kannada, Tulu words are frequently misclassified. Class imbalance also skews predictions towards the more frequent TULU tag. A curated list of orthographic or morphological patterns unique to each language could help improve separability.



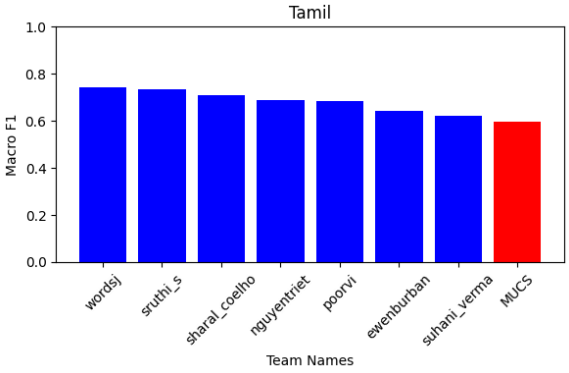
(a) Kannada – 6th Rank



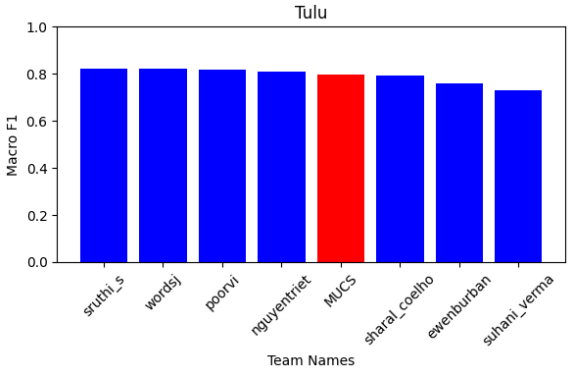
(b) Malayalam – 6th Rank



(c) Telugu – 4th Rank



(d) Tamil– 8th Rank

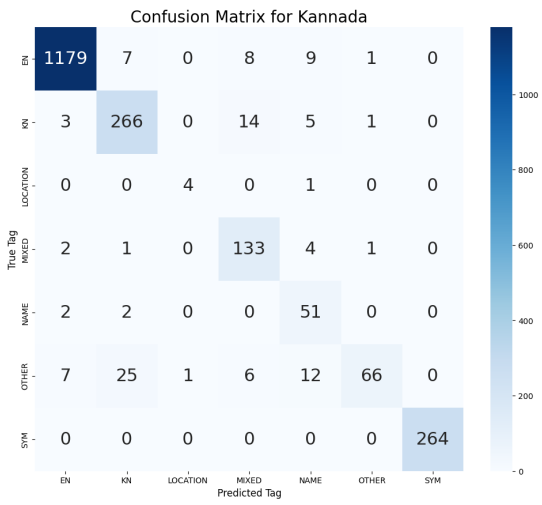


(e) Tulu – 5th Rank

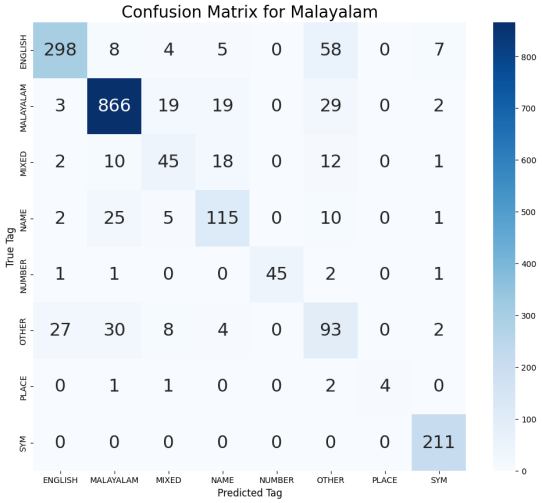
Figure 2: Language-wise Performances of all the Participating Teams in CoLI-Dravidian@FIRE 2025 Shared Task

While each language presents its own set of challenges, several recurring error patterns are consistent across all five languages. These issues arise primarily from multilingual interference, limited representation of rare tags, and orthographic ambiguities in transliterated or borrowed words. For example, i) MIXED class emerges as the most error-prone due to heterogeneity and ii) orthographically ambiguous or low-frequency tags (e.g., LOCATION, NUMBER) are also challenging, although strong orthographic cues enable better separation in tags like SYM. Such trends indicate that certain error types stem from structural and distributional properties of the data rather than language-specific phenomena.

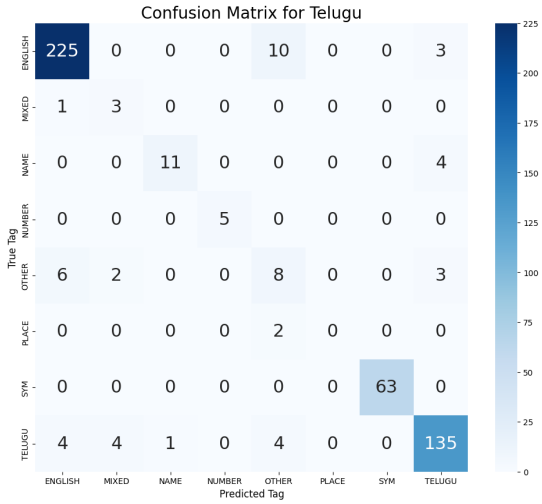
A significant challenge to the performance of the proposed models is data imbalance. Training sets of all the languages are substantially imbalanced. For example, tags like ENGLISH and TAMIL dominate, while low-resource tags such as MIXED, KANNADA, and LOCATION are underrepresented.



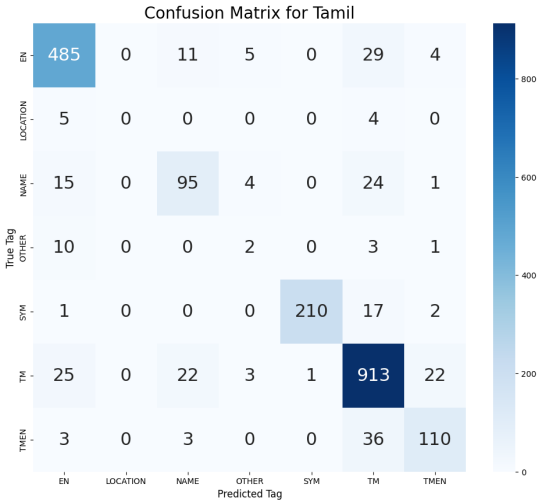
(a) Kannada



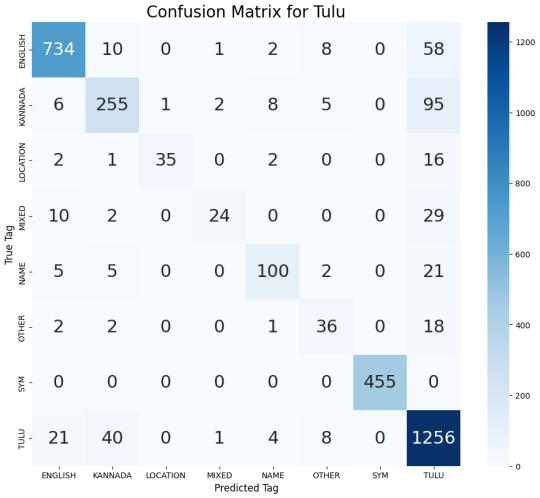
(b) Malayalam



(c) Telugu



(d) Tamil



(e) Tulu

Figure 3: Confusion Matrices Showing Class-wise Predictions for the Five Languages

The early experimentations with class reweighting and minority upsampling increased sensitivity on rare tags at the cost of significant overfitting and reduced macro-F1 performance. Taking this into considering and based on the CRF-based model developed by Asha et al. [11], we opted to use the imbalance Train sets provided by the organizers of the shared task. Nonetheless, improving error rates on low-frequency tags remains a key direction. Advanced imbalance-aware training (e.g., focal loss, synthetic data augmentation, or curriculum sampling) could further reduce confusion in future iterations.

4.3. Ablation Study

We explored a **Naive Baseline** that serves as a simple benchmark for sequence labeling tasks. It completely ignores the input text and assigns the most frequent label from the training set to every token in the test set, regardless of its actual content or context. For example, if *KANNADA* is the most common tag in the training data, then every word in the test set will be labeled as *KANNADA*, irrespective of its true language or identity. This serves as a sanity check and establishes a lower bound for performance comparison.

The features used to train the proposed CRF-based word-level LI are categorized into three major groups: (i) contextual word window features, (ii) character-level prefix and suffix n-grams, and (iii) word shape features (e.g., capitalization, presence of digits, special characters). To understand the contribution of different feature sets in our proposed model, We performed an ablation study by systematically removing one feature group at a time. The results show that removing **Prefix and suffix n-gram features** causes the largest drop in performance, highlighting the importance of morphological cues in handling agglutinative Dravidian languages. **Word shape features** contribute modestly, offering small but consistent gains. In contrast, removing **contextual word features** improves the performance for all languages, suggesting that the CRF's sequential modeling is effective in capturing local dependencies from the **Word shape** and **Prefix and suffix N-gram** features. Overall, this analysis confirms that the success of our CRF model stems not only from the algorithm itself but also from the inclusion of well-designed, language-specific, and morphology-aware feature engineering.

The results of the ablation study and the **Naive Baseline** in terms of F1 score are presented in Table 6. Ablation study reveals that the following consistent patterns emerge across the five Dravidian languages:

- **Prefix and Suffix N-grams:** This feature group proves to be the most influential. Removing it leads to a substantial drop in F1 score—up to **13% in Tamil** and over **6% in Kannada**—indicating that subword morphological patterns play a crucial role, particularly in agglutinative languages.
- **Contextual Word Features:** Surprisingly, excluding these features results in little to no performance degradation; in fact, slight improvements are observed for most languages. This suggests that the CRF's sequential modeling and lexical features are sufficient to capture local dependencies, making this group less critical in our setup.
- **Word Shape Features:** These contribute modestly, with minimal variation (less than 1% difference) when omitted. Their utility appears somewhat language-dependent—slightly more beneficial for Malayalam and Tamil—likely due to inconsistent capitalization and informal orthography common in code-mixed digital text.
- **Naive Baseline:** The naive model performs poorly across all languages (average F1 $\approx$ 31%), reaffirming the difficulty of the task and the advantages of engineered linguistic features and structured CRF modeling.

Overall, these findings confirm that the effectiveness of our CRF-based system is heavily dependent on well-designed linguistic feature engineering—particularly character n-gram morphology—which is especially crucial in code-mixed, low-resource scenarios.

Table 6
Performances of Proposed Feature Ablation Study in terms of F1 Score (%)

Feature Setting	Kannada	Malayalam	Telugu	Tamil	Tulu
Naive Baseline	28.70	25.69	43.12	33.78	24.46
With All Features	90.40	76.20	72.89	59.55	79.63
Without Contextual Word Features	91.00	77.34	74.97	62.45	80.92
Without Prefix/Suffix N-grams	83.56	69.77	72.22	46.67	73.99
Without Word Shape Features	90.00	76.08	73.70	58.55	79.73

5. Declaration on Generative AI

While drafting this paper, the authors utilized AI-based tools such as grammar correction and formatting support to assist in improving clarity and presentation. All core ideas, experimental design, implementation, interpretation of results, and written content were solely developed and curated by the authors. The final submission reflects original human-authored work grounded in independent research and critical analysis.

6. Conclusion and Future Work

In this paper, we presented our approach for the CoLI-Dravidian shared task at FIRE 2025, focusing on fine-grained word-level LI in code-mixed multilingual social media texts involving five Dravidian languages - Kannada, Malayalam, Tamil, Telugu and, Tulu. Through a CRF-based pipeline and carefully engineered features, our system demonstrated strong performance for Kannada and Tulu, while highlighting the inherent challenges in handling Tamil due to script ambiguity and overlapping vocabulary with English. Despite encouraging results, several limitations persist. The system occasionally struggles with ambiguous words, named entities, and transliterated words particularly in noisy informal text. These challenges point to the need for more context-aware and deep semantic models. This study lays a foundation for deeper exploration into multilingual and code-mixed language processing within the Dravidian language family, with potential applications in conversational AI, social media moderation, and regional language technologies. We would like to explore transformer-based multilingual models and character-level embeddings to better capture contextual dependencies. Incorporating external linguistic resources or pretraining on larger domain-specific corpora could also help to improve the performance, especially for low-resource languages like Tulu. Moreover, a focus on semi-supervised or zero-shot methods may further extend the scalability of our system to unseen dialects and languages.

References

[1] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, K. Lindén, Automatic Language Identification in Texts: A Survey, Journal of Artificial Intelligence Research 65 (2019) 675–782.

[2] M. Lui, V. Zue, J. Glass, Automatic Language Identification for Transcribed Speech, Interspeech (2014).

[3] K. G. Sridhar, A Survey of Code-Mixed Data and Approaches in Natural Language Processing, arXiv preprint arXiv:2004.00245 (2020).

[4] 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.

[5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural Architectures for Named Entity Recognition, in: Proceedings of NAACL-HLT, 2016, pp. 260–270.

[6] D. Kondratyuk, UFAL Submission to the IWPT 2019 Shared Task: Parsers for 50 Languages using 50 Treebanks, Proceedings of the Shared Task at the 15th International Conference on Parsing Technologies (IWPT) (2019).

[7] 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.

[8] H. L. Shashirekha, F. Balouchzahi, M. D. Anusha, G. Sidorov, Coli-Machine Learning Approaches for Code-mixed Language Identification at the Word Level in Kannada-English texts, Acta Polytechnica Hungarica 19 (2022) 123–141.

[9] A. Hegde, F. Balouchzahi, S. Coelho, S. HL, 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, 2023, pp. 25–26.

[10] 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: FIRE (Working Notes), 2023, pp. 179–190.

[11] 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), ACM, Gandhinagar, India, 2024, pp. 38–41. URL: https://doi.org/10.1145/3734947.3735663. doi:10.1145/3734947.3735663.

[12] B. R. Chakravarthi, R. Priyadharshini, M. A. Kumar, P. Krishnamurthy, E. Sherly, DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, 2021, pp. 1–11.

[13] R. Shimi, R. Thomas, S. Rajeev, An Empirical Evaluation of Machine Learning and Transformer-Based Models for Code-Mixed Text Classification, Journal of Computational Linguistics and Applications (2024). To appear.

[14] M. Gaman, T. Jauhiainen, M. Lui, M. Zampieri, Findings of the Vardial Evaluation Campaign 2021, in: Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2021), Association for Computational Linguistics, 2021, pp. 1–15.

[15] N. Deroy, S. Maity, Prompting GPT-3.5 for Word-Level Language Identification in South Indian Code-Mixed Texts, ArXiv preprint (2024). ArXiv:2403.10258.

[16] A. Hande, P. Yogi, B. R. Chakravarthi, J. P. McCrae, Offensive Language Identification in Dravidian Code-Mixed Social Media Texts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, 2021, pp. 18–26.

[17] V. Mandalam, N. Sharma, Sentiment Analysis on Code-Mixed Dravidian Languages using TF-IDF and Deep Learning Approaches, in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, CEUR-WS.org, India, 2021, pp. 123–130.

[18] G. Saumya, R. Rajeev, R. Thomas, Tanglish and Manglish Offensive Content Detection: A Comparative Study of Classical and Neural Approaches, in: Proceedings of the DravidianLangTech-EACL 2021 Workshop, Association for Computational Linguistics, 2021, pp. 132–138.

[19] S. Bose, G. Kharola, S. K. Naskar, IndicNLP@KGP at DravidianLangTech-EACL2021: Ensembles of Transformer and LSTM models for Offensive Language Identification in Code-Mixed Texts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, 2021, pp. 27–35.