Word-level Language Identification in Code-mixed Tulu Texts

Sushma N, Asha Hegde and Hosahalli Lakshmaiah Shashirekha

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

Abstract

Word-level Language Identification (LI) is the task of identifying the language of every word within a given multilingual sentence as in the case of code-mixed text. It is an essential pre-processing step for various language dependent applications such as machine translation. Though several research works are available for word-level LI in high-resource languages like Spanish, and French in multilingual context, many under-resourced languages are not yet explored in this direction. "CoLI-Tunglish: Word-level Language Identification in Code-mixed Tulu Texts" shared task organized at Forum for Information Retrieval Evaluation (FIRE) 2023, invites researchers to develop models to address the challenges of word-level LI in Tulu - an under-resourced Dravidian language. In this paper, we - team MUCS, describe the learning models submitted to this shared task for word-level LI in Tulu. Two distinct models: CoLI-Ensemble - an ensemble of Machine Learning (ML) classifiers (Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR)) with hard voting trained using character n-grams in the range (1, 3) and fastText pre-trained word vectors individually, and CoLI-CRF - a Conditional Random Field (CRF) algorithm trained with text-based features, are proposed for word-level LI in code-mixed Tulu text. Among the proposed models, CoLI-CRF outperformed the other model with a macro F1-score of 0.77 securing 4th rank in the shared task.

Keywords

Language identification, Tulu, Sequence labeling, Machine learning, Word embeddings

1. Introduction

In multilingual country like India, people are proficient in more than one language and often express themselves using a combination of two or more languages on social media platforms like Twitter, Instagram, Facebook, etc., [1, 2]. This mixture of languages, known as Code-mixing, involves the mixing of words or sub-words of more than one language at the word, phrase, or sentence level, with either a single script or multiple scripts [3]. Despite the availability of various applications that enable entering data in local/native languages, users frequently opt to use Roman script due to the technical limitations of computer keyboards and smart phone keypads, to key in Indian language characters, and the ease of using Roman script for transforming information in a convenient way [4, 5]. This has made Code-mixing a common phenomena especially on social media platforms.

Processing code-mixed text is challenging as it needs the tools/models that could handle multiple languages and multiple scripts in a given text [6]. The majority of the available

Forum for Information Retrieval Evaluation, December 15-18, 2023, India

sush.prgm@gmail.com (S. N); hegdekasha@gmail.com (A. Hegde); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

computational tools and pre-trained models, however, can only support monolingual text, highlighting the demand for effective tools and models to handle code-mixed text. Further, lack of digital resources for code-mixed text adds another dimension to the challenges associated with processing code-mixed text.

Tulu is an under-resourced language that belongs to the Dravidian language family and is spoken by more than three million people in the coastal regions of Karnataka and Karnataka-Kerala border. People who have considered Tulu as their mother tongue are known as Tuluvas and they are also found in Mumbai, Maharashtra, and many Gulf countries. Tulu language contains several Kannada words and as Tulu script is not popular, people commonly use Kannada script to write Tulu text. Further, people specifically those who are active on social media platforms, use either Kannada or Roman scripts or a combination of both to post their comments/reviews resulting in code-mixed text.

Models for Natural Language Processing (NLP) tasks like Machine Translation (MT) [7], Transliteration [8], Parts-Of-Speech (POS) tagging [9], Named Entity Recognition [10], etc., are conventionally designed for monolingual text. Using such models for code-mixed text directly may result in the degraded performance of these models, due to diverse linguistic structures of code-mixed text. This emphasizes the importance of language detection to ensure the quality of the applications/algorithms for processing code-mixed text which is multilingual in nature.

The preliminary step in processing code-mixed text is to identify the language of each word in a given sentence [11]. Word-level LI in high-resource languages like French and Spanish, and under-resourced languages like Hindi, Bengali, Tamil, Telugu, Kannada, and Malayalam, have been explored by many researchers [12, 13, 14, 15]. However, Tulu has never been explored in this direction, due to the non-availability of datasets and computational tools for this language. To address the challenges of word-level LI in Tulu, in this paper, we - team MUCS, describe the learning models submitted to "CoLI-Tunglish: Word-level Language Identification in Code-mixed Tulu Texts" shared task organized at FIRE 2023. The aim of this shared task is to develop learning models to tag one of the six classes, viz., Tulu, English, Kannada, Mixed, Name, Location, and Other, to each word in a given sentence. This shared task is modeled as a sequence labeling problem with two distinct models: i) CoLI-Ensemble - an ensemble of ML classifiers (SVM, RF, and LR) trained separately with Term Frequency-Inverse Document Frequency (TF-IDF) of character n-grams in the range (1, 3) and fastText word embeddings and ii) CoLI-CRF - a CRF classifier trained with text-based features, to identify the language of each word.

The rest of the paper is organized as follows: Section 2 contains Related Work and Section 3 describes the Methodology. While Section 4 gives the description of the Experiments and Results, the paper concludes with future work in Section 5.

2. Related work

Code-mixing in the context of Indian languages has become the default language of social media and it has attracted considerable research interest in word-level LI with several notable works [16]. The following description provides an overview of few Word-level LI works relevant to the study:

Shashirekha et al. [15] created a dataset for word-level LI in code-mixed Kannada text with 19,432 unique words and also collected code-mixed Kannada text with 72,815 unique sentences to build pre-trained models. The authors implemented four distinct models: i) CoLI-ngrams - an ensemble of three ML classifiers (LR, Linear Support Vector Classifier (Linear SVC), Multilayer Perceptron (MLP)) with soft voting trained with count vectors of character n-grams obtained from sub-word tokens, ii) CoLI-vectors - a pre-trained embeddings created considering words, sub-words, and characters from code-mixed Kannada text and used to train both ML and Deep Learning (DL) models iii) CoLI-BiLSTM - a DL model trained with CoLI-vectors, and iv) CoLI-ULMFiT - a Universal Language Model Fine-Tuning (ULMFiT) model pre-trained on raw text and is fine-tuned with the Train set, for word-level LI in code-mixed Kannada text. Among the models, CoLI-ngrams model obtained a macro F1-score of 0.64. A code-mixed Telugu-English dataset with 29,503 tokens is created by Gundapu and Mamidi [14] for word-level LI. To benchmark their dataset, the authors trained Naïve Bayes (NB) and RF classifiers, with TF-IDF of characters sequences, and Hidden Markov Model (HMM) and CRF models with text-based features. Among these models, CRF model obtained a macro F1-score of 0.91.

Thara and Poornachandran [13] created an annotated corpus of 7,75,430 tokens for word-level LI in code-mixed Malayalam-English text and implemented a wide range of transformer-based models (Bidirectional Encoder Representations from Transformers (BERT), distilled version of BERT (DistilBERT), Enhanced Light Efficiency Cophasing Telescope Resolution Actuator (ELECTRA), Cross-lingual Language Model Robustly Optimized BERT Approach (XLMRoberta), and CamemBERT). Among their proposed models, ELECTRA model obtained a macro F1-score of 0.9933. Mandal and Singh [17] proposed a novel approach for word-level LI in code-mixed Bangla and Hindi texts. Their proposed methodology has two phases: i) implementing Multichannel Neural Networks (MNN) by combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) and ii) feeding the output of MNN to Bidirectional LSTM+CRF model. With this approach, they obtained macro F1-scores of 93.49 and 93.32 for code-mixed Bangla and Hindi texts respectively.

Veena et al. [12] developed word embeddings as a function of the character embeddings of the characters present in the word, for word-level LI in code-mixed Tamil and Malayalam texts. They also used word embeddings of word trigrams and 5-grams as context features for each word. By training two individual SVM models for each context feature and word embeddings, the SVM model trained with 5-grams context features achieved macro F1-scores of 91.52 and 94.77 for code-mixed Malayalam and Tamil texts respectively. Barman et al. [18] created a trilingual (Bengali, English and Hindi) code-mixed dataset with 26,475 tokens for word-level LI. To benchmark their dataset, the authors trained SVM with TF-IDF of character n-grams in the range n = (1, 5) and CRF model with text-based features. Their proposed CRF model outperformed the other model with an accuracy of 95.76%.

From the available literature, it is clear that researchers have explored character n-grams, character embeddings, and BERT models to train conventional ML models and the NN models, for word-level LI in different Dravidian languages. To the best of our knowledge, word-level LI in code-mixed Tulu text has not been explored so far and Tulu is an under-resourced Dravidian language. This gives ample scope to explore various algorithms for word-level LI in code-mixed Tulu text.

3. Methodology

The proposed methodology for word-level LI in Tulu code-mixed texts consists of two models: i) CoLI-Ensemble and ii) CoLI-CRF. Pre-processing the dataset is not required as the dataset provided by the shared task organizers is clean and ready to use. The proposed models are described below:

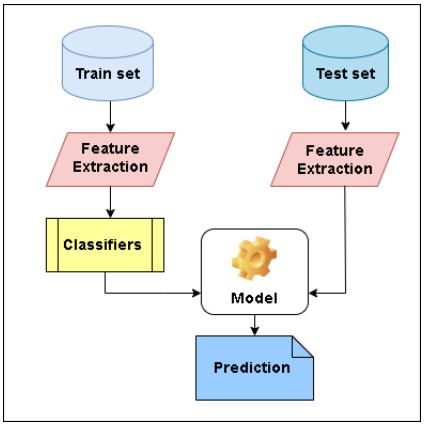

Figure 1: Framework of the proposed methodology

3.1. CoLI-Ensemble

This model consists of feature extraction followed by classifier construction. Description of each of these steps are given below:

3.1.1. Feature Extraction

Features play a significant role in deciding the performance of a classifier and the aim of feature extraction is to extract distinguishable features that can be used to train the learning models. CoLI-Ensemble models makes use of the following features:

- **Character n-grams** - is a sequence of 'n' characters in a romanized word. As it captures the structure of words, it can be conveniently used to represent the words in any romanized code-mixed texts. In this work, character n-grams in the range (1, 3) are extracted and vectorized using TFIDFVectorizer[1] to get the TF-IDF representation.
- **Pre-trained word embeddings** - are vector representation of words computed using vast amount of text data in any language. These embeddings are language dependent and encapsulate both the meaning and structure of words, enabling them to encode semantic and syntactic nuances and relationships between words. The only pre-trained models available for Tulu are fastText embeddings[2] and Byte-Pair Encoding (BPEmb)[3]

[1]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[2]https://fasttext.cc/docs/en/pretrained-vectors.html
[3]https://bpemb.h-its.org/tcy/

and the vocabulary size of both these pre-trained are very small (Tulu fastText - 7,000 and BPEmb - 10,000). In this work, Tulu, Kannada (vocabulary size - 1,88,249) and English (vocabulary size - 20,00,000) fastText pre-trained word embeddings of size 300 are used. The strength of the fastText pre-trained model is its capability to handle sub-word information, particularly well-suited for languages with rich morphological structures. This strength arises from its use of character n-grams, which enables it to represent and understand words even when they share sub-word components with other words.
Transliteration is a process of converting text from one script or writing system to another, preserving the pronunciation or sound of the original text rather than its meaning. As the pre-trained models are language dependent and the given dataset is in Roman script, all the words are transliterated to Kannada script (it may be noted that Tulu is written in Kannada script) using Libindic[4] library. With this arrangement, the words in the given dataset are available both in Kannada and Roman scripts. The following procedure is used to extract the word embeddings:

- If the word (either in Kannada and Roman script) is present in the vocabulary of any one (Kannada, Tulu and English) of the pre-trained models, the embeddings of the word is extracted from the corresponding pre-trained model.
- If the word is present in the vocabulary of more than one pre-trained models, then the embedding for that word is considered from the pre-trained model of the language to which it belongs in the dataset (ie., tag of that word). As many Kannada words are used in Tulu language, there is a chance that some words may be present in both Kannada and Tulu vocabularies of the pre-trained models. Similarly many English words may be present in Kannada/Tulu vocabularies of the pre-trained models.
- If the word is not present in the vocabularies of any of the above three pre-trained models, such words are considered as Out-Of-Vocabulary words. The embeddings of such words is created as a aggregation of character embeddings of the characters present in a word in Roman script and English fastText embeddings is used for this purpose.

The feature vectors which are obtained from the above feature extraction methods are then used to train the ensemble of ML classifiers.

3.1.2. Classifier Construction

Ensemble model is a method of generating a new classifier from multiple diverse base classifiers taking advantage of the strength of one classifier to overcome the weakness of another classifier with the intention of getting better performance for the classification task [19]. This arrangement of more than one diverse classifiers is guaranteed to outperform the constituent classifiers in the ensemble when considered individually. In ensemble models, several classifiers work together by voting to predict the class label of a sample. The proposed CoLI-Ensemble model ensembles three ML classifiers (SVM, LR, and RF) with hard voting. Description of the classifiers used in this model is given below:

[4]https://github.com/libindic/indic-trans

Table 1
Hyperparameters and their values used in CoLI-Ensemble model

Model Name	Hyperparameter and values
SVM	class_weight='balanced'
RF	n_estimators=100, max_depth=None, n_jobs=-1
LR	-

- **SVM** - is an ML classifier primarily designed for binary classification tasks To apply SVM to multiclass classification, common strategies like One-vs-Rest and One-vs-One are employed. One-vs-Rest trains multiple binary classifiers, one for each class, while One-vs-One trains pairwise classifiers for all possible class combinations, allowing SVM to effectively handle multiclass classification by reducing it to a series of binary decisions.
- **RF** - is an ensemble learning method that constructs multiple decision trees during training. Each tree is built independently and then their predictions are combined to yield a more accurate and robust overall prediction. By aggregating the outputs of numerous individual trees, RF reduces overfitting and enhances the model's performance [20].
- **LR** - is a ML algorithm specifically designed for binary classification tasks. Similar to SVM, multi-class classification in LR is approached through one-vs-rest scheme, where separate binary classifiers are trained for each class.

Hyperparameters and their values used to train SVM, RF, and LR in the CoLI-Ensemble model are given in Table 1 and default values are used for rest of the hyperparameters.

3.2. CoLI-CRF

Given the sequences of observations (words) in a sentence, CRF models the conditional probability distribution of tags (e.g., POS tags and Named Entity (NE) tags). CRF's strength lies in its ability to capture dependencies between tags considering both the preceding and succeeding observations, allowing it to make context-aware predictions in tagging tasks (e.g. POS tags and NE tags) [21]. For large and structured tag sets, CRF works well with many features that are mutually dependent. In this work, CRFSuite is implemented using sklearn_crfsuite[5] library, which acts as a wrapper for CRF implementation. This library simplifies the classifier construction process by wrapping the transformation of textual features into feature vectors and training the CRF classifier. Features used to train the CRF classifier in the proposed CoLI-CRF model are shown in Table 2.

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

Table 2
Features used in CoLI-CRF model

Features	
A word	Previous word-2
Length of word	Previous word-3
Is the word at the beginning of the sentence	Previous word-4
Is the word at the end of the sentence	Next word+2
Is current word digit	Next word+3
Is current word punctuation	Next word+4

Table 3
Class-wise distribution of CoLI-Tunglish dataset

Category	# of Comments
Tulu	8,647
English	5,499
Kannada	2,068
Name	1,104
Other	506
Mixed	403
Location	369

Table 4
Sample words and the corresponding labels in CoLI-Tunglish dataset

Category	Description	Samples
Name	Name of a person	shivam, ayyapa
Location	Indicates the location	kudla, udupi
English	Pure English words	Sir, super
Tulu	Tulu words in Roman script	Apundu, pura
Kannada	Kannada words in Roman script	visaya, anna
Mixed	Combination of Kannada, Tulu and/ or English	vedion, photoga
Other	Words not belonging to any categories	git, mujhe

4. Experiments and Results

The CoLI-Tunglish dataset contains code-mixing of three languages (Tulu, Kannada, and English) in Roman script for the purpose of word-level LI and consists of seven labels (Tulu, Kannada, English, Mixed, Name, Location, and Other). Label distribution of the CoLI-Tunglish dataset is given in Table 3 and the sample words from the dataset and the corresponding labels followed by their descriptions are given in Table 4.

Table 5
Performance of the proposed models

Model	Features	Development set			Test set		
		Precision	Recall	F1-score	Precision	Recall	F1-score
CoLI-Ensemble	Character n-grams	0.86	0.65	0.69	0.86	0.57	0.63
	fastText word embeddings	0.85	0.84	0.83	0.87	0.69	0.75
CoLI-CRF	Text features	0.86	0.84	0.87	0.80	0.74	**0.77**

Several experiments were conducted with various feature sets (Tulu BPEmb, OOV embeddings, and a combination of these embeddings with Tulu fastText embeddings and textual features) to train a wide range of ML classifiers (SVM, LR, RF, k-NN, MLP, DT, and CRF). Models that exhibited better performances for the Development set are evaluated on the Test set and the performance of the proposed models for the Development and the Test sets are shown in Table 5. The results indicate that CoLI-CRF model has exhibited better macro F1-score than the CoLI-Ensemble model due to the ability of CRF model to capture the context. However, CoLI-Ensemble model trained with the feature vectors extracted from the fastText pre-trained word embeddings has exhibited slightly lesser macro F1-score than that of CoLI-CRF model due to the small vocabulary size of Tulu.

5. Conclusion

This paper describes the models submitted by our team - MUCS, to "CoLI-Tunglish: Word-level Language Identification in Code-mixed Tulu Texts" shared task at FIRE 2023, for word-level LI in code-mixed Tulu texts. Two distinct models: i) CoLI-Ensemble - a model that adopts ensembling of ML classifiers (SVM, RF, and LR) trained with TF-IDF of character n-grams in the range (1, 3) and fastText word embeddings separately and ii) CoLI-CRF - a CRF model trained with text-based features, are proposed for word-level LI in code-mixed Tulu text. Among the proposed models, CoLI-CRF model achieved a macro F1-score of 0.77 for word-level LI in code-mixed Tulu text securing 4^{th} rank in the shared task.

References

[1] C. M. Scotton, The Possibility of Code-Switching: Motivation for Maintaining Multilingualism, in: Anthropological linguistics, JSTOR, 1982, pp. 432–444.

[2] A. Hegde, H. L. Shashirekha, Leveraging Dynamic Meta Embedding for Sentiment Analysis and Detection of Homophobic/Transphobic Content in Code-mixed Dravidian Languages, 2022.

[3] S. H. Lakshmaiah, F. Balouchzahi, M. D. Anusha, G. Sidorov, CoLI-Machine Learning

Approaches for Code-Mixed Language Identification at the Word Level in Kannada-English Texts, in: Acta Polytechnica Hungarica, 10, 2022, pp. 123–141.
[4] A. Hande, R. Priyadharshini, B. R. Chakravarthi, KanCMD: Kannada CodeMixed Dataset for Sentiment Analysis and Offensive Language Detection, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, 2020, pp. 54–63.
[5] F. Balouchzahi, H. Shashirekha, LA-SACo: A Study of Learning Approaches for Sentiments Analysis in Code-Mixing Texts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 109–118.
[6] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text, in: Language Resources and Evaluation, Springer, 2022, pp. 765–806.
[7] A. Hegde, S. Lakshmaiah, Mucs@ mixmt: Indictrans-based Machine Translation for Hinglish Text, in: Proceedings of the Seventh Conference on Machine Translation (WMT), 2022, pp. 1131–1135.
[8] D. K. Sharma, A. Singh, A. Saroha, Language Identification for Hindi Language Transliterated Text in Roman Script using Generative Adversarial Networks, in: Towards Extensible and Adaptable Methods in Computing, Springer, 2018, pp. 267–279.
[9] K. Ball, D. Garrette, Part-of-speech Tagging for Code-switched, Transliterated Texts without Explicit Language Identification, in: Proceedings of the 2018 conference on empirical methods in natural language processing, 2018, pp. 3084–3089.
[10] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named Entity Recognition for Code-mixed Indian Corpus using Meta Embedding, in: 2020 6th international conference on advanced computing and communication systems (ICACCS), IEEE, 2020, pp. 68–72.
[11] D. Nguyen, A. S. Doğruöz, Word Level Language Identification in Online Multilingual Communication, in: Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 857–862.
[12] P. Veena, M. A. Kumar, K. Soman, An Effective way of Word-Level Language Identification for Code-Mixed facebook Comments using Word-Embedding via Character-Embedding, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 1552–1556.
[13] S. Thara, P. Poornachandran, Transformer based Language Identification for Malayalam-English Code-Mixed Text, in: IEEE Access, IEEE, 2021, pp. 118837–118850.
[14] S. Gundapu, R. Mamidi, Word Level Language Identification in English Telugu Code Mixed Data, in: arXiv preprint arXiv:2010.04482, 2020.
[15] 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, in: arXiv preprint arXiv:2211.09847, 2022.
[16] A. Jamatia, A. Das, B. Gambäck, Deep Learning-based Language Identification in English-Hindi-Bengali Code-mixed Social Media Corpora, De Gruyter, 2019, pp. 399–408.
[17] S. Mandal, A. K. Singh, Language Identification in Code-Mixed Data using Multichannel Neural Networks and Context Capture, in: arXiv preprint arXiv:1808.07118, 2018.

[18] U. Barman, A. Das, J. Wagner, J. Foster, Code Mixing: A Challenge for Language Identification in the Language of Social Media, in: Proceedings of the first workshop on computational approaches to code switching, 2014, pp. 13–23.
[19] A. Hegde, H. L. Shashirekha, Urdu Fake News Detection Using Ensemble of Machine Learning Models, in: CEUR Workshop Proceedings, 2021, pp. 132–141.
[20] H. Jhamtani, S. K. Bhogi, V. Raychoudhury, Word-Level Language Identification in Bi-Lingual Code-Switched Texts, in: Proceedings of the 28th Pacific Asia Conference on language, information and computing, 2014, pp. 348–357.
[21] Machine Learning Approaches for Amharic Parts-of-Speech Tagging, in: arXiv preprint arXiv:2001.03324, 2020.