Word-Level Language Identification in Dravidian Code-Mixed Text Using Machine Learning: A Comparative Analysis of Models and Vectorization Techniques

Ponsubash Raj R, Bharathi B

Department of Computer Science, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India

Abstract

Language Identification (LI) is a major component for various applications such as Sentiment Analysis, Machine Translation, Information Retrieval, and Natural Language Processing. In multilingual countries like India, especially among the younger generation, social media often contains code-mixed texts where local languages are combined with English. This presents significant challenges for LI, particularly at the word level, where languages may be mixed even within a single word. Dravidian languages, which are widely spoken in southern India, are morphologically rich but under-resourced, making word-level LI in these languages even more complex. This paper presents a Machine Learning approach to tackle word-level LI in four Dravidian languages: Tamil, Kannada, Tulu, and Malayalam, each of which is combined with English in the datasets. The study employs vectorization techniques such as Count Vectorizer and TF-IDF Vectorizer, alongside models like Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier, and a Voting Classifier. Our findings highlight the importance of selecting the appropriate model and vectorization technique based on the specific linguistic context. For instance, while the Voting Classifier with TF-IDF Vectorization achieved the highest overall macro F1-scores in certain languages, Logistic Regression with Count Vectorization demonstrated superior performance in language-specific classifications, particularly in identifying mixed words. Additionally, the paper addresses potential dataset biases and discusses future directions to expand the model's applicability across diverse and underrepresented languages.

Keywords

Word-Level Classification, Language Identification, Code-mixed texts, Dravidian Languages, Machine Learning

1. Introduction

India, with its rich cultural and linguistic diversity, is a country where multilingualism is not just common but a way of life. People often express themselves using a combination of languages, especially on social media platforms, where it is becoming increasingly popular to mix languages in posts and comments. It's a particularly prevalent trend among younger generations who effortlessly switch between languages to convey their thoughts and emotions.

However, this linguistic flexibility also presents a unique challenge for language processing tasks. Language mixing can occur at multiple levels, across entire paragraphs, within sentences, at the word level, or even within individual words, where elements from different languages are combined. For example, a word might originate from one language but include prefixes or suffixes from another, adding to the complexity. This fluidity of language use makes it difficult to accurately identify and process the languages involved.

Most existing computational tools and pre-trained models are designed for monolingual text and they struggle to handle the intricacies of code-mixed content, which often involves multiple languages and scripts within a single text. The lack of sufficient digital resources for code-mixed text further complicates the task, highlighting the need for specialized tools and models that can manage this linguistic diversity. Accurate language identification at the word level is a crucial first step in understanding and analyzing code-mixed text for various applications.

In response to these challenges, this paper explores word-level language identification (LI) for Dravidian languages, focusing on the "CoLI-Dravidian: Word-level Code-Mixed Language Identification

Forum for Information Retrieval Evaluation, December 12-15, 2024, India

ponsubashraj2370043@ssn.edu.in (P. R. R); bharathib@ssn.edu.in (B. B)

in Dravidian Languages" shared task, part of FIRE 2024[1]. The study evaluates the performance of five machine learning models—Multinomial Naïve Bayes (MNB), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree Classifier (DTC), and a Voting Classifier ensemble of LR, SVM, and DTC—on datasets that include Tamil, Kannada, Tulu, and Malayalam, each mixed with English. The effectiveness of these models is analyzed using two distinct vectorization techniques: Count Vectorizer and Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. The datasets, which are a combination of one Dravidian language and English, are further detailed in Section 4 of this paper.

This paper aims to provide insights into how different models and vectorization techniques perform in the context of multilingual, code-mixed text, for the development of more effective tools for language processing in diverse linguistic environments. While this paper focuses on a select set of Dravidian languages (Tamil, Kannada, Tulu, and Malayalam), it lays the groundwork for further research in Word-Level Language Identification (LI) across other underrepresented languages and dialects. Expanding this approach to a broader set of languages, including less common code-mixed scenarios, could provide valuable insights for multilingual natural language processing tasks.

2. Related Work

Asha Hegde et al. [2] conducted a comprehensive evaluation of traditional machine learning algorithms for word-level language identification (LI) in code-mixed Tulu texts. Their approach utilized Term Frequency-Inverse Document Frequency (TF-IDF) representations of bi-grams and tri-grams to capture the linguistic patterns inherent in the code-mixed data.

H.L. Shashirekha, F. Balouchzahi, M.D. Anusha, and G. Sidorov [3], along with Lakshmaiah [4], conducted a detailed study on word-level language identification (LI) for Kannada-English (Kn-En) code-mixed texts. Their research incorporated both traditional machine learning and deep learning techniques, introducing models like CoLI-ngrams and CoLI-vectors for machine learning, as well as CoLI-BiLSTM and CoLI-ULMFiT for deep learning. One of the primary challenges they encountered was the absence of an annotated Kn-En code-mixed dataset. To address this, they developed the CoLI-Kenglish dataset by scraping comments from Kannada YouTube videos and manually tagging them. Native Kannada speakers annotated the data, classifying it into six linguistic categories.

Asha Hegde et al. [5] provided an in-depth overview of the methodologies and outcomes of the "CoLI-Tunglish: Word-level Language Identification in Code-mixed Tulu Texts" shared task. This task featured participation from five distinct teams, each employing various strategies for word-level language identification (LI) in code-mixed Tulu texts. Among the approaches, a machine learning model that employed a stacking ensemble of multiple classifiers trained on character n-grams emerged as the top performer. This model achieved a significant macro F1 score of 0.813, highlighting its effectiveness in addressing the complexities of code-mixed Tulu text processing[6].

In the context of word-level language identification in code-mixed Kannada-English texts, particularly during the ICON 2022 competition, a notable trend emerged. As documented by Balouchzahi et al. [7], teams that leveraged neural network (NN) architectures and transformer-based models consistently outperformed traditional machine learning classifiers and baseline models.

Thara and Poornachandran[8] developed an extensive annotated corpus of 775,430 tokens specifically for word-level language identification (LI) in code-mixed Malayalam-English texts. They evaluated the effectiveness of several transformer-based models, including BERT, DistilBERT, ELECTRA, XLM-Roberta, and CamemBERT, in processing and identifying the languages within this code-mixed dataset.

Veena and colleagues[9] introduced a method using word embeddings based on character embeddings within words for word-level LI in code-mixed Tamil and Malayalam texts. They also explored the use of word embeddings derived from word trigrams and 5-grams as contextual features. By training Support Vector Machine (SVM) models on these features, they achieved high macro F1-scores of 91.52 and 94.77 for code-mixed Malayalam and Tamil texts, respectively.

Barman et al.[10] compiled a trilingual code-mixed dataset of Bengali, English, and Hindi, containing 26,475 tokens for word-level LI. They established a baseline using an SVM model trained on TF-IDF

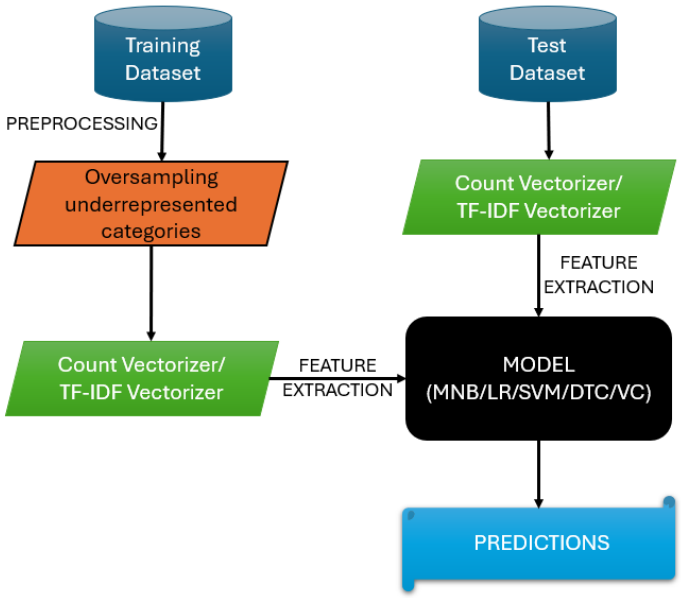

Figure 1: Framework of Proposed Methodology

of character n-grams and a Conditional Random Fields (CRF) model with text-based features. The CRF model outperformed the SVM, achieving an accuracy of 95.76%. Their earlier work (2014)[11] highlighted the importance of character n-grams and contextual information for language identification in Indian language code-mixing.

In the track, Sentiment Analysis for Dravidian Languages in Code-Mixed Text (FIRE 2020)[12], while the top-ranked submissions used BERT-based models for Tamil in the Dravidian Languages in Code-Mixed Text track, another approach using word embedding modeling achieved a weighted F1 score equal to that of the highest-scoring team.

K. K. Akhil, R. Rajimol, and V. S. Anoop (2020)[13] demonstrated that deep learning-based methods outperform some state-of-the-art techniques in language computing tasks, particularly in parts-of-speech tagging for Indic languages. Mandal and Singh (2018)[14] introduced a novel architecture for language tagging of code-mixed data, which effectively captures contextual information, enhancing the accuracy of language identification in such complex datasets.

To provide a clear structure for the paper, Section 3 presents a detailed explanation of the proposed models and methodologies employed for Word-Level LI in code-mixed Dravidian text. Section 4 outlines the experimental setup, including dataset preparation, model training, and evaluation results. Finally, Section 5 offers an in-depth analysis of the experimental outcomes, highlighting key insights and comparing the performance of different models and vectorization techniques.

3. Proposed Model

3.1. Methodology

This paper analyzes the performances of 5 models using 2 different vectorizers for the Word-Level Language Identification of Dravidian Languages. The methodology is shown in Figure 1.

3.2. Datasets and Pre-processing

The contents of the four available datasets from the shared task are as follows:

- Tulu: The Tulu dataset contains 7,171 code-mixed sentences scraped from YouTube videos, pre-processed to remove non-text characters and transliterated into Roman script. From these, 36,002

Table 1
Datasets

Language	Training Set	Validation Set
Tamil	22091	1984
Kannada	48680	2484
Tulu	43948	3006
Malayalam	45399	2504

words are categorized into six classes: 'Tulu', 'Kannada', 'English', 'Mixed-language', 'Name', and 'Location', posing challenges due to the dynamic and individualistic nature of mixed-language words.

- Kannada: This dataset includes 14,847 tokens in Roman script, classified into six categories: 'Kannada', 'English', 'Mixed-language', 'Name', 'Location', and 'Other'. It aims to improve language identification and categorization methods for Kannada-English code-mixed texts.
- Tamil: Comprising 17,568 tokens, the Tamil dataset follows a similar methodology to Tulu and Kannada datasets, categorized into six classes. It supports various natural language processing tasks within the Tamil language domain.
- Malayalam: The Malayalam dataset contains 25,035 tokens, divided into six classes: 'Malayalam', 'English', 'Mixed', 'Name', 'Number', and 'Location'. It offers comprehensive coverage for NLP tasks, similar to the other datasets, but includes additional categories like 'Number' for numerical values.

There is an extra 'sym' category for each of the datasets, to indicate the sentence boundary.

With the current dataset, there is a huge disparity in the available examples for different categories. For example, in the Tulu dataset, there are over 11,000 words with the tag 'Tulu' and over 7,000 words with the tag 'English', but there are only about 500 words with the tag 'Mixed'. This affects the model predictions, giving very less probabilities for such low-data categories and biased towards the categories with more data. To address this issue, we have oversampled the categories with lesser amount of data. In this way, the model can learn more about the less resourced categories and give an overall better F1-score.

Specifically, for the Malayalam dataset, the numbers in the dataset are preprocessed to words, to better represent them during vectorization of the word. Further, to improve the identification of 'Name' and 'Location' categories in each dataset, they are retrieved from all the language datasets and used in common, as the name and location identification should not depend on the language. This increases the data trained for each language and also improves the F1-score for those categories.

It is important to note that the dataset used in this study, while diverse, may still reflect certain biases inherent to social media language usage. The informal nature of online communication, including slang, abbreviations, and cultural references, may limit the model's applicability to formal or more structured code-mixed data. Future research will explore the impact of these biases by incorporating datasets from formal contexts, such as news articles or academic texts, to achieve a more balanced evaluation.

3.3. Vectorizers Used

We have compared and analyzed the performance of 2 Vectorizers in the Word-Level Language Identification of Dravidian Languages.

Count Vectorizer: The Count Vectorizer was employed to convert text into numerical features by extracting character n-grams ranging from unigrams (1-grams) to 5-grams (5-grams). The analyzer was set to 'char', which means the vectorizer breaks down the text at the character level.

TF-IDF Vectorizer: The TF-IDF Vectorizer was also utilized with character n-grams in the range of 1 to 5 and the analyzer set to 'char'. This vectorizer not only considers the frequency of character sequences but also adjusts for the importance of these sequences by down-weighting those that appear frequently across different documents.

Table 2
Model Performance on Tamil-English validation dataset

Model Name	Accuracy	Recall	Macro F1 Score	Weighted F1 Score
MNB (Count Vectorizer)	0.84	0.59	0.68	0.84
LR (Count Vectorizer)	0.92	0.79	0.69	0.92
DTC (Count Vectorizer)	0.87	0.74	0.63	0.88
SVM (Count Vectorizer)	0.91	0.63	0.64	0.92
Voting Classifier (Count Vectorizer)	0.92	0.78	0.69	0.92
MNB (TF-IDF Vectorizer)	0.87	0.57	0.59	0.87
LR (TF-IDF Vectorizer)	0.91	0.62	0.63	0.91
DTC (TF-IDF Vectorizer)	0.86	0.72	0.62	0.86
SVM (TF-IDF Vectorizer)	0.91	0.75	0.69	0.91
Voting Classifier (TF-IDF Vectorizer)	0.91	**0.76**	**0.70**	0.91

3.4. ML Models Used

We have compared and analyzed the performance of 5 different ML Models in the Word-Level Language Identification of Dravidian Languages.

Multinomial Naive Bayes: Multinomial Naive Bayes classifiers were employed for discrete feature classification tasks, utilizing both Count Vectorizer and TF-IDF Vectorizer. These models, particularly well-suited for text data, operate under the assumption that features (words or terms) follow a multinomial distribution.

Logistic Regression: Logistic Regression models were implemented using the newton-cg solver, optimized for large datasets. To ensure convergence, especially in the context of complex datasets, the models were configured to run for a maximum of 2000 iterations. Both Count Vectorizer and TF-IDF Vectorizer were used to process the data for these models.

Support Vector Machine: Support Vector Machines (SVM) with a linear kernel (svm.SVC()) were applied. SVMs are particularly effective in high-dimensional spaces and are well-suited for scenarios where the number of features exceeds the number of samples, making them ideal for the text classification tasks in this study.

Decision Tree Classifier: Decision Tree Classifiers were utilized with a fixed random state of 42 to ensure reproducibility. These models are non-parametric, known for their interpretability, and are capable of handling both categorical and continuous data, making them versatile tools for the classification tasks at hand.

Voting Classifier: A Voting Classifier was constructed to harness the complementary strengths of the Decision Tree Classifier, Logistic Regression, and SVM models. The classifier employed a hard voting strategy, which determines the final class label based on the majority vote from the constituent models.

4. Experimental Results

4.1. Tamil

In the Word-Level LI of the Tamil-English dataset, the Voting Classifier model using TF-IDF Vectorization had the best macro F1-score of 0.70, on the validation dataset. It was closely followed by the models, LR model using Count Vectorizer, SVM using TF-IDF Vectorization and Voting Classifier model using Count Vectorizer having a macro F1-score of 0.69.

The LR model using Count Vectorization excelled in specific categories, delivering the best macro F1-scores for Tamil, English, and mixed (Tamil-English combined) categories. This suggests that while the Voting Classifier with TF-IDF Vectorization was the top performer overall, the LR model with Count Vectorization may be more effective for certain language-specific classifications within the dataset.

Table 3
Model Performance on Kannada-English validation dataset

Model Name	Accuracy	Recall	Macro F1 Score	Weighted F1 Score
MNB (Count Vectorizer)	0.87	0.75	0.73	0.87
LR (Count Vectorizer)	**0.95**	**0.83**	**0.85**	**0.95**
DTC (Count Vectorizer)	0.92	0.79	0.80	0.92
SVM (Count Vectorizer)	0.95	0.79	0.83	0.94
Voting Classifier (Count Vectorizer)	0.95	0.81	0.84	0.95
MNB (TF-IDF Vectorizer)	0.92	0.67	0.68	0.91
LR (TF-IDF Vectorizer)	0.94	0.79	0.82	0.94
DTC (TF-IDF Vectorizer)	0.92	0.81	0.81	0.92
SVM (TF-IDF Vectorizer)	**0.95**	0.81	**0.85**	**0.95**
Voting Classifier (TF-IDF Vectorizer)	0.95	0.81	0.84	0.95

4.2. Kannada

In the Word-Level LI of the Kannada-English validation dataset, the SVM model using TF-IDF Vectorizer had the best score, along with LR model using Count Vectorizer, with a Macro F1 Score of 0.85 and accuracy of 0.95. Even though both the models had similar scores at the categorical level, the LR model using Count Vectorizer performed slightly better in categorizing mixed words(Kannada and English combined words) with an F1 score of 0.92, compared to 0.88 for the SVM model using TF-IDF Vectorizer. This suggests that frequency-based vectorization techniques like Count Vectorizer may better capture the nuances of mixed-language words, possibly due to its ability to emphasize the frequency of word components.

These findings highlight the importance of considering both the model and the vectorization technique when working with bilingual or multilingual datasets. Although SVM with TF-IDF is highly effective overall, Logistic Regression with Count Vectorizer might offer better performance in specific linguistic scenarios, particularly when dealing with mixed words.

4.3. Tulu

In the Word-Level LI of Tulu-English validation dataset, most of the models had comparable performances. The models, SVM using Count Vectorizer and Voting Classifier using Count Vectorizer had the best Macro F1 Score of 0.83. They are closely followed by the LR model using Count Vectorizer, Voting Classifier using TF-IDF Vectorizer, and the SVM model using TF-IDF Vectorizer, having a Macro F1 Score of 0.82. The LR model had the best Macro F1 Score for the categories, Tulu, and English with scores 0.93 and 0.92 respectively, and the second-best Macro F1 Score for the categories Kannada and mixed(Tulu-English combined), both with scores 0.75.

The close performance of different models and vectorization techniques suggests that multiple approaches are viable for Word-Level LI in the Tulu-English context. However, the LR model's strong performance in individual categories, particularly with Count Vectorization, indicates its reliability in accurately identifying specific languages within the dataset.

4.4. Malayalam

In the Word-Level LI of the Malayalam-English validation dataset, the Voting Classifier model using Count Vectorizer had the highest Macro F1 Score of 0.87, followed by the LR model using Count Vectorizer and the SVM model using Count Vectorizer and Voting Classifier model using TF-IDF Vectorizer, with a Macro F1 Score of 0.85. In this dataset, even though the LR model using TF-IDF Vectorizer got the third highest F1 Score of 0.84, it had the highest Macro F1 Score for the categories English, Malayalam, and mixed(Malayalam-English combined) with the scores 0.92, 0.96 and 0.64 respectively.

Table 4
Model Performance on Tulu-English validation dataset

Model Name	Accuracy	Recall	Macro F1 Score	Weighted F1 Score
MNB (Count Vectorizer)	0.80	0.77	0.69	0.81
LR (Count Vectorizer)	0.90	0.83	0.82	0.90
DTC (Count Vectorizer)	0.86	0.78	0.77	0.86
SVM (Count Vectorizer)	0.89	0.82	**0.83**	0.89
Voting Classifier (Count Vectorizer)	0.90	0.82	**0.83**	0.90
MNB (TF-IDF Vectorizer)	0.85	0.75	0.75	0.84
LR (TF-IDF Vectorizer)	0.88	0.81	0.81	0.88
DTC (TF-IDF Vectorizer)	0.86	0.79	0.76	0.86
SVM (TF-IDF Vectorizer)	0.90	0.81	0.82	0.90
Voting Classifier (TF-IDF Vectorizer)	0.89	0.81	0.82	0.89

Table 5
Model Performance on Malayalam-English validation dataset

Model Name	Accuracy	Recall	Macro F1 Score	Weighted F1 Score
MNB (Count Vectorizer)	0.85	0.78	0.76	0.85
LR (Count Vectorizer)	0.91	0.84	0.84	0.91
DTC (Count Vectorizer)	0.90	0.86	0.82	0.90
SVM (Count Vectorizer)	0.92	0.82	0.85	0.92
Voting Classifier (Count Vectorizer)	**0.93**	0.85	**0.87**	**0.92**
MNB (TF-IDF Vectorizer)	0.88	0.70	0.72	0.87
LR (TF-IDF Vectorizer)	0.91	0.81	0.83	0.91
DTC (TF-IDF Vectorizer)	0.90	0.84	0.80	0.90
SVM (TF-IDF Vectorizer)	0.92	0.83	0.84	0.92
Voting Classifier (TF-IDF Vectorizer)	0.92	0.83	0.85	0.92

The analysis of the Word-Level Language Identification (LI) for the Malayalam-English validation dataset reveals that the Voting Classifier model using Count Vectorizer achieved the highest overall performance with a Macro F1 Score of 0.87. This indicates that the ensemble approach, which combines the strengths of multiple models, is particularly effective in handling the diverse linguistic features of the Malayalam-English dataset. While the Voting Classifier using Count Vectorizer emerged as the most effective model overall, the LR model using TF-IDF Vectorizer stands out for its category-specific performance, making it particularly useful for tasks that require precise language classification at the word level. This highlights the importance of selecting models based on the specific requirements of the task, whether it be overall accuracy or performance within particular language categories.

5. Result Analysis

5.1. Vectorization

Count Vectorization consistently outperformed or matched TF-IDF Vectorization across multiple datasets, particularly in the Tamil-English and Kannada-English datasets. This suggests that frequency-based word representations are highly effective in capturing the nuances of multilingual data, especially when dealing with mixed-language words. The ability of Count Vectorization to capture n-grams and word frequency patterns likely contributes to its superior performance in these scenarios.

5.2. Model Performance

Logistic Regression (LR) and Support Vector Machine (SVM) emerged as strong contenders across the datasets, often achieving the highest or near-highest macro F1 scores. LR, in particular, showed

Table 6
Test Data Performance

Language	M-F1	M-Pr	M-Re	W-F1	W-Pr	W-Re	acc
Tamil	0.766	0.76	0.788	0.917	0.919	0.917	0.917
Kannada	0.852	0.872	0.858	0.937	0.943	0.938	0.938
Malayalam	0.876	0.919	0.85	0.923	0.926	0.926	0.926
Tulu	0.816	0.843	0.799	0.895	0.896	0.896	0.896

remarkable performance in specific language categories, making it a reliable choice for detailed language identification tasks. SVM, with its effectiveness in high-dimensional spaces, also proved to be a robust model, particularly when paired with the appropriate vectorization technique. The Voting Classifier, which combined the strengths of LR, SVM, and DTC models, demonstrated the highest overall performance in several datasets, particularly in the Malayalam-English context. This underscores the value of ensemble methods in leveraging the complementary strengths of individual models to improve overall classification accuracy.

6. Conclusion

The comprehensive analysis of Word-Level LI across the four datasets—Tamil-English, Kannada-English, Tulu-English, and Malayalam-English—reveals that both model choice and vectorization technique play critical roles in achieving high classification accuracy. Count Vectorization generally proved more effective across these multilingual datasets, particularly in capturing the nuances of mixed-language words. Logistic Regression and Support Vector Machines emerged as the most reliable models, with the Voting Classifier demonstrating the highest overall performance in several cases.

This study highlights the importance of tailoring machine learning approaches to the specific linguistic challenges presented by multilingual datasets. By carefully selecting the appropriate models and vectorization techniques, it is possible to achieve high levels of accuracy and robustness in Word-Level Language Identification tasks, even in complex, mixed-language scenarios.

A key limitation of our study is the potential bias introduced by the social media-based dataset. Informal language, including regional slang and cultural references, may impact the model's performance in real-world applications, especially in more formal or non-social media contexts. Future work should focus on curating datasets that represent both informal and formal use cases, to ensure the robustness of the model across different domains.

The model will be extended to include more languages and dialects in future research to evaluate its generalization to different linguistic contexts. This will involve creating and testing datasets that reflect various linguistic structures and code-mixed environments beyond Dravidian languages.

We have submitted the Voting Classifier model using Count Vectorizer in the final submission of the test data prediction for the shared task and got 1st place in the Tamil-English dataset, 5th place in Kannada-English dataset, 2nd place in the Malayalam-English dataset and 4th place in the Tulu-English dataset.

7. Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Hegde, F. Balouchzahi, S. Butt, S. Coelho, K. G, H. S Kumar, S. D, S. Hosahalli Lakshmaiah, A. Agrawal, Overview of CoLI-Dravidian: Word-level Code-mixed Language Identification in Dravidian Languages, in: Forum for Information Retrieval Evaluation FIRE - 2024, 2024.

[2] 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, 2022, pp. 33–40.
[3] S. Hosahalli Lakshmaiah, F. Balouchzahi, A. Mudoor Devadas, G. Sidorov, CoLI-Machine Learning Approaches for Code-mixed Language Identification at the Word Level in Kannada-English Texts, acta polytechnica hungarica (2022).
[4] 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, Acta Polytechnica Hungarica 19 (2022).
[5] 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, 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.
[6] A. Hegde, F. Balouchzahi, S. Coelho, H. 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.
[7] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, S. Hosahalli Lakshmaiah, G. Sidorov, A. Gelbukh, Overview of CoLI-Kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at ICON 2022, in: 19th International Conference on Natural Language Processing Proceedings, 2022.
[8] S. Thara, P. Poornachandran, Transformer based language identification for malayalam-english code-mixed text, IEEE Access 9 (2021) 118837–118850.
[9] 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.
[10] U. Barman, J. Wagner, J. Foster, Part-of-speech tagging of code-mixed social media content: Pipeline, stacking and joint modelling, in: Proceedings of the second workshop on computational approaches to code switching, 2016, pp. 30–39.
[11] 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.
[12] A. V. Mandalam, Y. Sharma, Sentiment analysis of dravidian code mixed data, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, 2021, pp. 46–54.
[13] K. Akhil, R. Rajimol, V. Anoop, Parts-of-speech tagging for malayalam using deep learning techniques, International Journal of Information Technology 12 (2020) 741–748.
[14] S. Mandal, A. K. Singh, Language identification in code-mixed data using multichannel neural networks and context capture, arXiv preprint arXiv:1808.07118 (2018).