INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

An Empirical Analysis of Language Detection in Dravidian Languages

G Shimi[1], **C Jerin Mahibha**[2]*, **Durairaj Thenmozhi**[3]

1 Department of Computer Applications, Madras Christian College, Tambaram, Chennai, 600059, Tamil Nadu, India
2 Department of Computer Science and Engineering, Meenakshi Sundararajan Engineering College, Kodambakkam, Chennai, 600 024, Tamil Nadu, India
3 Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, 603 110, Tamil Nadu, India

OPEN ACCESS

Received: 01-04-2023

Accepted: 13-03-2024

Published: 04-04-2024

Citation: Shimi G, Mahibha CJ, Thenmozhi D (2024) An Empirical Analysis of Language Detection in Dravidian Languages. Indian Journal of Science and Technology 17(15): 1515-1526. https://doi.org/10.17485/IJST/v17i15.765

* **Corresponding author**.

jerinmahibha@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Language detection is the process of identifying a language associated with a text. The proposed system aims to detect the Dravidian language that is associated with the given text using different machine learning and deep learning algorithms. The paper presents an empirical analysis of the results obtained using the different models. It also aims to evaluate the performance of a language agnostic model for the purpose of language detection. **Method:** An empirical analysis of Dravidian language identification in social media text using machine learning and deep learning approaches with k-fold cross validation has been implemented. The identification of Dravidian languages, including Tamil, Malayalam, Tamil Code Mix, and Malayalam Code Mix, is performed using both machine learning (ML) and deep learning algorithms. The machine learning algorithms used for language detection are Naive Bayes (NB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), and Random Forest (RF). The supervised Deep Learning (DL) models used include BERT, mBERT and language agnostic models. **Findings:** The language agnostic model outperform all other models considering the task of language detection in Dravidian languages. The results of both the ML and DL models are analyzed empirically with performance measures like accuracy, precision, recall, and f1-score. The accuracy associated with different machine learning algorithms varies from 85% to 89%. It is evident from the experimental result that the deep learning model outperformed with an accuracy of 98%. **Novelty:** The proposed system emphasizes on the use of the language agnostic model to implement the process of detecting Dravidian languages associated with the given text which provides a promising result of 98% accuracy which is higher than the existing methodologies.

Keywords: Language; Machine learning; Deep learning; Transformer model; Encoder; Decoder

1 Introduction

Different native languages prevail in different parts of the world, each with its own script, symbols, and syntax. India is a country with an ancient and morphologically huge diversity of native languages[(1)]. The people living in India also belong to a multilingual community in which sharing information in code-mixed languages is common. Code-mix language is the semantic usage of expressions using different languages[(2)]. The usage of language has changed incredibly throughout the world under the influence of social media. This makes the detection of the language associated with the posted information an important area of research in the field of natural language processing. Various tools that work on multilingual data need to detect the language associated with the text automatically. Language detection is relevant to computational linguistics, where it could be considered one of the most challenging tasks. Linguistic research in low resource languages like the Dravidian languages requires language detection to take the research to a higher level. Effective automated solutions for hate speech detection and sentiment analysis have been implemented using different machine learning and deep learning methods[(3)]. Sentiment analysis and offensive language identification for low resource code-mixed data in Tamil and English had been implemented using machine learning, deep learning and pre-trained models like BERT, RoBERTa and adapter-BERT[(4)]. Transliteration of English to Tamil Unicode characters had been implemented using phonetics based forward list processing[(5)] by Anbukkarasi, S. et al. A combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) had been used to recognize the text in natural images[(6)].

English is the approved communicative language on most social media platforms[(7)]. People from multilingual countries like India prefer to mix their native language with English for better communication[(8)]. The code mix languages can be found in reviews, YouTube comments, feedback, articles, quotes, findings, etc. Humans can easily detect the languages that they are familiar with. Due to the large number of languages prevalent in India, an Indian himself may not be in a position to identify all the languages[(9)]. The language identification of the texts is hard since the labelling of these texts is not easy. This can be associated with different multilingual applications in which the accuracy and speed of detection play a major role[(10)]. Proper training is required for several people to be associated with several language identification services, but a language identification system that is trained once can be used simultaneously on different machines so that they support multiple services. The research question associated with the language detection task are:

(i) Is a gold standard dataset available for language detection task in Dravidian language?

(ii) Has promising results obtained by existing research considering Dravidian language detection task?

(iii) What techniques are explored to handle the language detection in Dravidian languages?

(iv) Whether a monolingual, multilingual or crosslingual model will provide better results for language detection task considering code mixed languages?

Language identification on a code-mixed dataset with text representing Hindi-English had been implemented using deep learning and transformer models, which used token classification to tokenize the input sentence[(11)]. Language and dialect identification tasks had been implemented using Naive Bayes classifiers with adaptive language models and a transformer-based model[(12)]. The Dravidian Language Identification (DLI) shared task dataset had been used for the implementation, which had code-mixed text with English and one of the three South Indian languages: Kannada, Malayalam, or Tamil. Competitive performance had been provided by the Naive Bayes model rather than the transformer model. Language identification of code-mixed text in monolingual language model pertained in English had been implemented using the transformer model BERT, which had been pretrained on a large amount of Hindi-Urdu-English code-mixed data. The RoBERTa transformer model had also been fine-tuned for downstream language detection[(13)]. Naive Bayes methods, Convolutional Neural Networks (CNN), and Deep Feedforward Neural Network (DNN) had been used to identify languages and dialects from text associated with one of the Italian language varieties[(14)]. Better performance had been provided by the Deep Feedforward Neural Network (DNN) when trained on character n-gram than by the other models.

As part of the VarDial Evaluation Campaign 2021, Yves Bestgen et al.[(15)] demonstrated the use of convolutional neural networks and shallow models for language identification. Comparable performance had been yielded by the models when applying data augmentation over the training dataset in the Romanian and the Dravidian Language Identification tasks. In the Uralic Language Identification task, an ensemble model built with Support Vector Machines and Naive Bayes performed better.

A dataset for sentiment analysis and offensive language identification had been developed for three under-resourced Dravidian languages from social media comments[(16)]. Language agnostic cross-lingual word embeddings had been used to detect hope speech in code-mixed Dravidian languages[(17)]. An ensemble of LSTM and BERT based transformer models to detect offensive language in code-mixed Dravidian languages[(18)] had been proposed by Kushal Kedia and Abhilash Nandy.

The details of the existing research in the area of language detection has been summarized in Table 1. The table shows the language considered, the methodology, and the dataset used for the research. One of the existing systems proposed by Andrea Ceolin[(14)] has an F1 score of 0.994, which considers Italian languages. The system proposed by Mohd Zeeshan Ansari et al.[(13)]

had used two models for implementing downstream language identification task and had achieved as F1score of 0.84. Subword vocabulary generation models of WordPiece and BLBPE had been utilized and word level classification had been carried out using RoBERTa. Considering Dravidian languages, an F1 score of 0.92[12] has been achieved on using the dataset provided by the shared task DLI@VarDial 2021 which had used Naïve Bayes classifier with character n- gram for implementing the process. The proposed Naïve Bayes' algorithm had outperformed the pretrained models. One of the reasons for this could be that the comments contained code-mixed sentences which the pretrained language models like BERT and XLM-R had not encountered before. The research gaps that exist considering Dravidian language detection task includes:

(i) Gold standard dataset for language detection considering Dravidian languages are not available.

(ii) Transformer based embedding schemes, are not much explored for the process of identifying Dravidian languages.

(iii) Cross lingual and language specific models have not been explored for detecting Dravidian languages.

(iv) Limited amount of research is available considering language detection in Dravidian language.

Table 1. Language Detection - Existing System

Literature	Language	Dataset	Methodology	Performance score
Tommi Jauhiainen et al.[12]	Tamil code-mix, Malayalam code-mix, Kannada code-mix	DLI@VarDial 2021	Naïve Bayes Classifier with character n-gram	Macro F1 score - 0.92
Mohd Zeeshan Ansari et al.[13]	Hindi code-mix	Custom dataset	RoBERTa model	F1 score - 0.84
Andrea Ceolin[14]	Italian languages	ITDI@VarDial 2022	Deep Feedforward Neural Networks	Macro F1 score - 0.99
Yves Bestgen[15]	Tamil code-mix, Malayalam code-mix, Kannada code-mix	DLI@VarDial 2021	Logistic Regression with character n-gram	Macro F1 score - 0.81

It could be observed that even though research is carried out for language detection, only limited research could be found considering Dravidian language detection. Low resource and dataset scarcity could be considered reasons for the limited amount of research in this area. It could also be noted that various machine learning algorithms are prevalently used for this research.

The proposed language identification system helps in identifying Dravidian languages of the Indian subcontinent, which include Tamil, code-mixed Tamil, Malayalam, and code-mixed Malayalam texts. Multi-class classification models considering both machine learning and deep learning models are used for the process of language detection to categorize the given text as Tamil, Tamil Code Mix, Malayalam, or Malayalam Code Mix. The proposed system employs a language agnostic model to implement the task. An empirical analysis for the same is carried out to show that the proposed language agnostic model outperforms other models using performance measures like accuracy, precision, recall, and f1-Score.

2 Methodology

The dataset that was used for detecting Dravidian languages is a custom generated dataset in which the text is in any one of the Dravidian languages considered, namely Tamil, Tamil code-mix, Malayalam and Malayalam code-mix. The proposed system uses a custom dataset as gold standard dataset for Dravidian language detection are not available. The dataset was constructed by combining the data provided as part of different shared tasks and Kaggle. The sentences are manually annotated as Tamil, Tamil code-mix, Malayalam, or Malayalam code-mix.

The custom dataset that was used to perform language detection based on multiclass classification models had 39010 instances with two features: text and the language associated with it. The text encompasses sentences from any one of the following languages: Tamil, Tamil code-mix, Malayalam, or Malayalam code-mix, which are annotated based on the language represented by the text. The number of instances in each category is tabulated in Table 2. There are 9787, 15335, 5036, and 8850 instances in the categories Tamil, Tamil code-mix, Malayalam, and Malayalam code-mix, respectively. Multiple languages including a regional language and English are often the choice of people to express their ideas and thoughts in social media platforms. This process is known as code-mixing, which refers to the embedding of linguistic units from one language into the usage of another language by using phrases, words, and morphemes.

Table 2. Dataset Description

Category	No. of Instances				
	Total	Tamil	Tamil Code Mix	Malayalam	Malayalam Code Mix
Training dataset	31206	7809	12284	4036	7077
Testing dataset	7802	1978	3051	1000	1773

The distribution of the data in the dataset is shown in Figure 1. It shows that the custom dataset has 25% of the instances in Tamil, 39% of instances in Tamil code mix, 13% of the instances in Malayalam and 23% of the instances in Malayalam code mixed language. This shows the imbalanced nature of the dataset with the major portion of the instances representing code mixed languages.

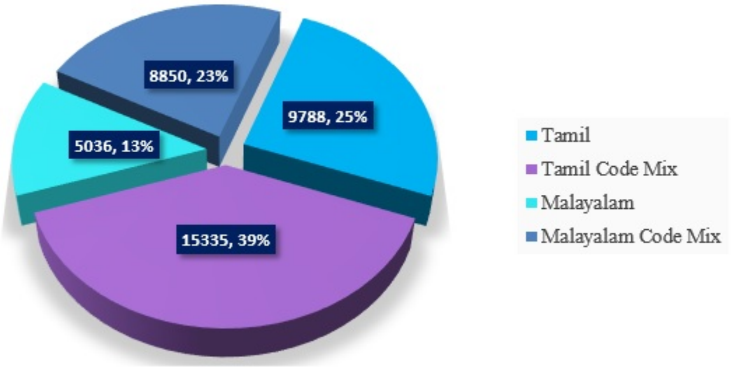

Fig 1. Distribution of data in Dataset

As the Dravidian languages like Tamil, Malayalam, and Kannada are closely related, with a few words being common in all these languages, the process of language detection is considered a challenging task[19]. The proposed system uses a language agnostic model to implement the language detection system. An empirical analysis of the system has also been implemented, using both traditional machine learning and transformer based models. The overall architecture of the proposed model is represented by Figure 2.

The input sentence written in the Dravidian languages is provided as input for the model. Pre-processing of the input is carried out to retain the useful parts, which is followed by classification of the input using both machine learning and transformer based models. An empirical analysis of the output obtained has also been conducted.

The traditional machine learning models, namely Naive Bayes (NB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), and Random Forest (RF), are used to implement the classification task of language detection. The transformer models used for the analysis include BERT, multi-lingual BERT and language-agnostic models.

Cross-validation is a repeated random subsampling method used to assess a predictive model's generalization ability and prevent overfitting, which can lead to higher model accuracy. As the dataset is a custom generated dataset, for better utilization of the available data, 5-fold cross validation is implemented in the proposed language detection system. By using a Label Encoder, the label representing the language is converted into multiclass labels. The unigram features of the text are extracted using the concept of vectorization, which converts the words from the sentence to corresponding vectors of real numbers, which help with word predictions in classification models[20].

2.1 Preprocessing

The cleaning of the data is a required attribute before applying the dataset to the predefined model. This is performed to remove noise from the dataset. The preprocessing step removes the noise, which is information that does not contribute to the language detection task and provides relevant information from the sentences before they are provided as input to the model. The preprocessing is done by removing the emoji, special symbols, hash tags, numbers, urls, and punctuation.

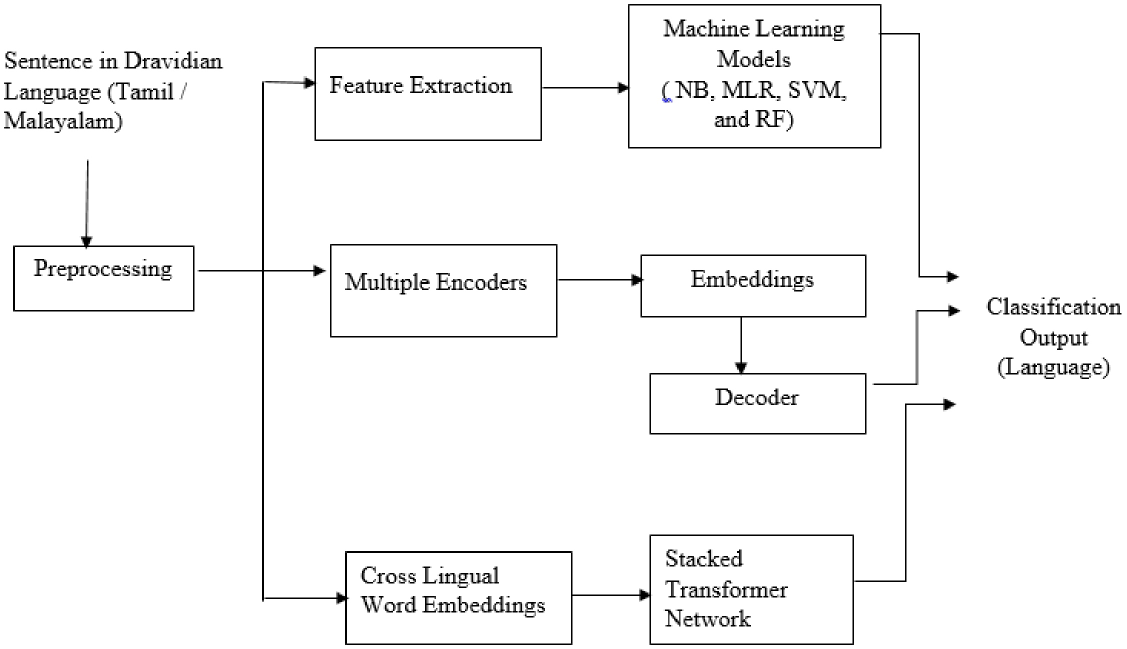
Sentence in Dravidian Language (Tamil / Malayalam)
Preprocessing
Feature Extraction
Machine Learning Models (NB, MLR, SVM, and RF)
Multiple Encoders
Embeddings
Decoder
Cross Lingual Word Embeddings
Stacked Transformer Network
Classification Output (Language)

Fig 2. Proposed Architecture

2.2 Machine Learning Models

A language detection system is a text classification problem for which various machine learning algorithms could be used. To detect the language associated with the text, the proposed system uses traditional machine learning algorithms like Naive Bayes (NB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), and Random Forest (RF). An empirical analysis of the algorithms is also carried out. The performance metrics associated with each of the above models are considered for the empirical analysis.

Naive Bayes[21] is a simple, efficient, and supervised machine learning model based on Bayes' theorem to solve classification problems, such as text classification. The probability of the text belonging to a particular language is computed using the mathematical equation given below, based on which the task of language detection is done.

$$P(Lang|f) = \frac{P(f|Lang) * P(Lang)}{P(f)}$$

The algorithm is implemented using the optimization function by setting the value of alpha to 1.0 and the flag fit-prior to True.

Logistic Regression[22] is the probabilistic based statistical model used to solve classification problems in machine learning. For both classification and regression problems, it can be used, but it is extensively used for classification problems[23]. Multinomial Logistic Regression is an extension of Logistic Regression used to perform multi-class classification. The probability distribution defines a multi-class probability[24]. The optimization parameters that are set for Multinomial Logistic Regression to implement language detection are solver as "lbfgs", maximum iteration as 5,000, balanced class weight and the tolerance for the stopping criteria as 1e-4.

Support vector machine[25] is a supervised machine learning model used for classification and regression analysis. The SVM performs both linear and nonlinear classification. The SVM optimization function uses scale as the value of the parameter gamma, rbf kernel, random state 100, balanced class weight, tolerance level of 1e-3 and cache size of 200 to detect the language.

Random Forest[26] is a tree structure based classifier in which the features are selected randomly in each decision split, which improves prediction power and efficiency. High dimensional data modelling can be performed by Random Forest since it can handle missing values and continuous, categorical, and binary data. Language detection is achieved with the number of estimators set to 100, random state as 100, balanced class weight, minimum sample split as 2, and sqrt as the maximum feature parameter value.

All machine learning algorithms provide a hyperparameter space using which the best combination of hyperparameters are identified to obtain optimal result for the language detection task. The two generic approaches used for this purpose includes Grid search cross validation and Random search cross validation techniques which are used for identifying the choice of hyper parameters for the proposed system. Irrelevant features, improper model parameters and imbalanced datasets are considered as some factors that contribute to poor accuracy of the system. Long Short-Term Memory (LSTM) had been used to classify fake and real news utilizing hyperparameter tuning methods such as grid search and random search to customize the hyperparameters of the model[27]. It could be found from the literature that for the process of implementing classification tasks, 3 to 5 epochs could be considered a better choice. During the process of parameter tuning, it was found that optimized results are generated with 3 epochs. While training the model beyond 3 epochs, it was observed that there was an increase in the value of the training loss. Hence, 3 epochs were selected for implementing different models by the proposed system.

During the process of parameter tuning, it was found that optimized results are generated with 3 epochs and hence the proposed system used 3 epochs for the process of implementing different models.

2.3 Transformer Models

To accomplish the language detection, from the different available transformer models, multilingual BERT and language agnostic models are chosen. The analysis is carried out by evaluating the metrics, namely accuracy, precision, recall, and F1-Score, using 5-fold cross validation. The model is trained for 3 epochs by setting the parameter representing the number of labels to 4. The BERT multilingual model is a pretrained model for 104 languages to predict unlabeled features. BERT is a case-sensitive transformer model that employs a self-supervised technique. The BERT makes use of a sequence of encoders, as represented in Figures 3 and 4 shows that a self-attention and feed forward network are part of each encoder.

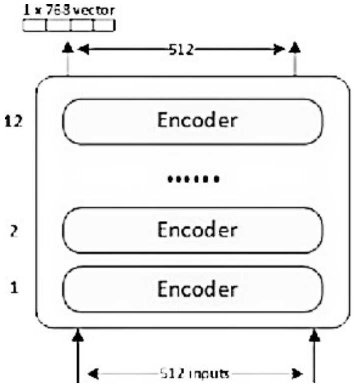

Fig 3. BERT Architecture

BERT is based on transformers, which is a deep learning model where every output element is connected to every input element, and dynamic computation of weight based on attention is done. Using the bidirectional capability of BERT, it is pre-trained on two related tasks, namely Masked Language Modelling and Next Sentence Prediction.

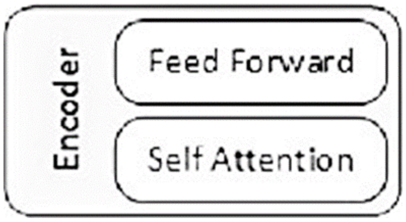

Fig 4. Encoder Structure - BERT

Transformer allows the BERT model to understand the full context of the word. The model can be fine-tuned by adding an output layer and trained by setting values to the hyperparameters. For implementing the language detection system, the model is tuned with the required parameters, like the number of labels being 4 and the number of epochs being 3. The optimizer used is Adam with a learning rate of 1e-4, $\beta 1$=0.9 and $\beta 2$=0.999, a weight decay of 0.01, learning rate warm up for 10,000 steps and linear decay of the learning rate. The flow of information in a BERT architecture is represented by Figure 5, which starts with the embedding layer generating the word embeddings (EN). A new intermediate representation (Trm) of fixed size is generated by every layer using the previous layer representation and multi-headed attention computation, and TN represents the final output.

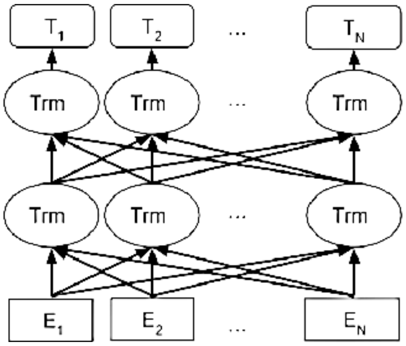

Fig 5. Information Flow in BERT

Multilingual BERT (mBERT)[28] is a transformer based neural language model, the architecture of which is the same as that of BERT-Base. The difference is that it uses a concatenation of monolingual Wikipedia corpus from 104 languages for pretraining the model. The pretrained model uses an embedding layer and 12 layers of transformer encoders for which sentences tokenized into a sequence of n tokens are provided as input. A sequence of contextual representations is generated for each token at each layer. During the entire training procedure, no explicit cross-lingual alignment is provided.

The BERT base model is supported by 12 intermediate layers. Language agnostic BERT Sentence Embedding[29] is a multilingual model for cross-lingual sentence embedding in 109 languages. Pre-training can be accomplished by combining masked language modelling (MLM) with translation language modelling (TLM). This model performs well in multilingual sentence embedding and multilingual text retrieval. To make the process of training more efficient, a dual-encoder architecture has been used, which is considered to be an effective approach for learning cross-lingual embeddings. The BERT transformer model forms the base of the encoder architecture, which has 12 transformer blocks, 12 attention heads, and 768 per-position hidden units. All languages share the various encoder parameters. Tokenization in Language agnostic model plays a crucial role in preparing the text for input into the model, allowing it to process and understand the semantic meaning of sentences across multiple languages. The input text is tokenized into smaller units using the WordPiece tokenizer. This involves breaking down words into sub word units. Each token is assigned a unique token ID, which corresponds to its index in the tokenizer's vocabulary. Special tokens are also added to the tokenized input to mark the beginning and end of sentences, as well as to denote padding or unknown tokens. Along with token IDs, attention masks are also generated to indicate which tokens are actual words and which ones are padding tokens. This helps the model focus only on the relevant parts of the input during processing. From the last transformer block, normalized [CLS] token representations are extracted as the sentence embeddings. As the model follows the bidirectional nature of encoders, the final embeddings are translations of each other, represented by s and t. The ranking of the true translation of s over all the sentences in T, containing the set of different sentences t$_i$, is carried out. The probability distribution for every t$_i$ given in a source text s is represented by the equation below.

$$P(t_i|s_i\ ) = \frac{e^{\varnothing(s_i,t_i)}}{\sum_{\bar{t}eT}\ e^{\varnothing(s_i,\bar{t})}}$$

An effective approximation of the probability distribution is achieved by training in-batch cross accelerated negative samples, as represented in the following equation:

$$P_{approx}(t_i|s_i) = \frac{e^{\varnothing(s_i,t_i)}}{e^{\varnothing(s_i,t_i)} + \sum_{k=1,k\neq i}^{k} e^{\varnothing(s_i,t_n)}}$$

A shared transformer network is used to encode the source and target text, and the translation ranking task helps to get similar representations for the source and target text. Mapping similar words from different languages to a common representation is part of the parameter sharing capacity of the encoders by altering the hyper parameters associated with the model, it is being trained. The pre-trained sentence embedding represents language independence as pair of sentences with same meaning and different languages are more similar when compared to pair of sentences with same language and different meaning. The model is trained with the objective of feature prediction. The number of labels is set to 4 while tuning the model for language detection, which is trained for 3 epochs. The model has been implemented with Adam optimizer and a batch size of 32. The process behind this method is represented by Figure 6.

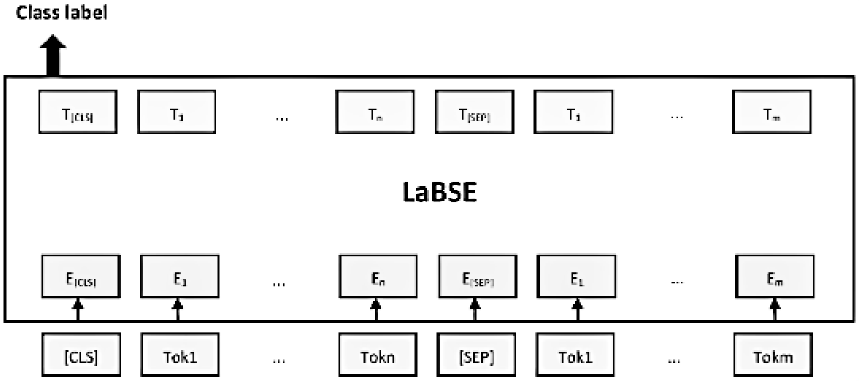

Fig 6. Language Agnostic BERT Model

Language embeddings represent entire languages as fixed-size vectors in an embedding space. These embeddings can capture various linguistic properties of languages, such as vocabulary, syntax, and semantics. They are often learned using large-scale multilingual corpora and techniques such as cross-lingual word embeddings or language modeling. In a language-agnostic model, the sentence embeddings that are generated capture the semantic meaning of the sentences, allowing for tasks like similarity comparison, classification, or translation across different languages without requiring language-specific processing. In the context of language detection tasks, language embeddings contribute by capturing the unique linguistic characteristics of different languages in a continuous vector space. Various machine learning models and transformer based models are used to implement the process of language detection, considering the Dravidian languages Tamil, Tamil code-mix, Malayalam, and Malayalam code-mix. The performance measures of all the models are analysed in the following section.

2.4 Empirical Analysis

The performance of the different machine learning and transformer based models is analysed by considering different performance metrics, which include accuracy, precision, recall, and f1-score. Accuracy represents the ratio of the number of correct predictions to the total number of input samples. The ratio of the number of correct positive results to the number of positive results predicted by the classifier is represented by precision. The model's ability to detect positive samples is represented by recall. The F1-score is an overall measure of a model's accuracy that combines precision and recall. A high f1-score means that the classification has resulted in a low number of false positives and false negatives. The scores of the above performance metrics obtained by Naive Bayes, MLR, SVM, and RF are given in Table 3.

Table 3. Performance

Model/performance metric	Accuracy	Precision	Recall	F1 score
Naive Bayes	85.19	0.87	0.86	0.86
MLR	88.19	0.90	0.90	0.89
SVM	88.69	0.90	0.89	0.89
RF	86.74	0.89	0.86	0.87

Table 4 tabulates the score of the performance metrics considering the transformer models, namely BERT and the language-agnostic model.

3 Results and Discussion

The experiment was carried out to detect language using different multiclass classification machine learning models, such as Naive Bayes (NB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), and Random Forest (RF), as well as transformer models, such as BERT, multilingual BERT and Language Agnostic Model, using the custom dataset. All the

Table 4. Performance

Model/Performance metric	Accuracy	Precision	Recall	F1 score
mBERT	98	0.98	0.97	0.98
BERT	92.5	0.90	0.90	0.92
Language Agnostic Model	**98.6**	**0.98**	**0.98**	**0.98**

implementation is accomplished using 5-fold cross validation of the dataset, and the mean of the metrics accuracy, precision, recall, and f1-Score are evaluated. Table 3 depicts a comparison of the performance metric scores of various machine learning models. It could be found that the SVM model outperforms all other models considering the performance metrics, namely accuracy, precision, recall, and F1-score. The performance metrics of the different machine learning models used for the process of language detection has been diagrammatically represented in Figure 7.

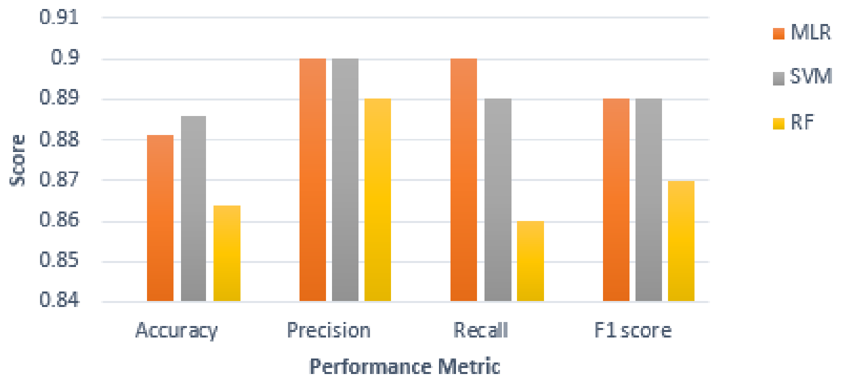

Fig 7. Performance Measures – Machine Learning Models

The resultant score of the different performance metrics associated with Transformer models is represented in Table 4. It is evident from the table that the language agnostic model provides a better result than the multilingual BERT model. But considering the metrics of precision and F1-score, both models have provided the same result. Figure 8 is the graphical representation of the performance scores of the Language detection model implemented using Transformer models.

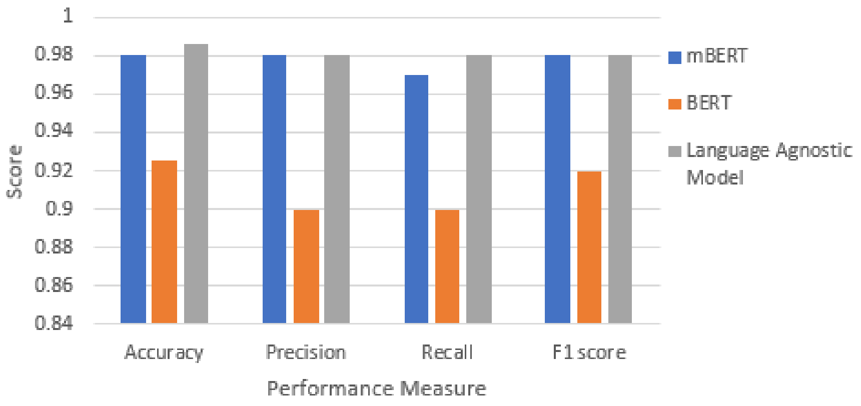

Fig 8. Performance Measures - Transformer Models

It is evident that the accuracy of the SVM model is 88.6% and that of the Language agnostic model is 98.6% for the task of language detection using the custom dataset. This shows that the language agnostic models provide better results than the traditional machine learning models on the language detection task using the custom generated dataset.

Model interpretability refers to the ability to understand and explain the decisions or predictions made by a machine learning model in a human-understandable manner. Interpretability is crucial for ensuring transparency, trustworthiness, and accountability in AI systems, especially in applications where decisions have significant consequences.

The performance metric of the proposed model could be compared with the methodologies used by different existing systems, which are tabulated in Table 5. The existing methodologies, namely Naïve Bayes Classifier and Logistic Regression with character n-gram were applied over the custom dataset and the accuracy obtained by them were 85.21 and 87.01 respectively The F1 score of the existing methodologies were 0.86 and 0.89 respectively. Considering Dravidian languages, the proposed model outperformed the existing models with a f1 score of 0.98 which is higher than the existing methodologies.

Table 5. Performance Scores using Existing Methodologies

Methodology	Accuracy	Precision	Recall	F1 score
Naïve Bayes Classifier with character n-gram [12]	85.21	0.87	0.86	0.86
RoBERTa [13]	97.8	0.98	0.97	0.97
Logistic Regression with character n-gram [15]	87.01	0.89	0.89	0.89
Language Agnostic Model	**98.6**	**0.98**	**0.98**	**0.98**

Ethical considerations related to language detection involve various aspects, including privacy, bias, fairness, and cultural sensitivity. Ensuring that language detection is developed and deployed ethically requires careful attention to these. By addressing these considerations thoughtfully, developers and users can harness the benefits of language detection while minimizing its potential risks and negative impacts.

The proposed methodology has been validated using a large dataset provided as part of the shared task ITDI@VarDial 2022 which was associated with Identification of Languages and Dialects of Italy (ITDI). The dataset had a total of 11085 instances with 8 languages to be identified. The languages include LIJ(Ligurian), LLD(Ladin), NAP(Neapolitan), FUR(Nilo-Saharan), VEC(Venetian), EML(Emilian-Romagnol), LMO(Lombard) and ROA_TARA(Romance). The distribution of the data in the dataset is represented by the Figure 9.

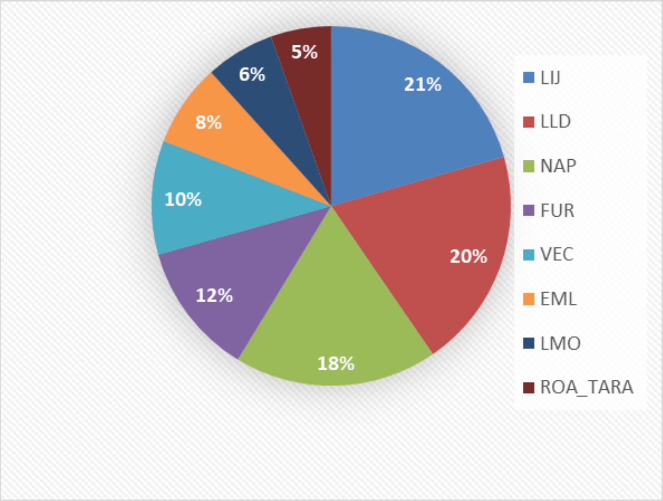

Fig 9. Data Distribution -ITDI@VarDial 2022

The proposed model when used to detect the languages represented by ITDI@VarDial 2022 dataset, has provided an Accuracy of 99.2% with a Precision of 0.99, Recall of 0.99 and F1-Score of 0.99. This shows that the proposed model provides an acceptable result even when used with larger datasets.

3.1 Error Analysis

When performing the task of language detection, there are sentences in all languages that are considered misclassified: Tamil, Malayalam, Tamil code mixed, and Malayalam code mixed. The lexical similarity that exists between these Dravidian languages can be considered a reason for the misclassification. The models were also not able to classify certain words, which could also be considered a reason for incorrect predictions. One of the major challenges with the code mixed data is the coexistence with other languages. Samples of misclassified sentences are provided in Table 6. From the table it could be observed that there are few words like thalivaa, aavum, vanam, chevantha which are part of both the languages Tamil and Malayalam, which could be considered as a reason for misclassification.

Table 6. Samples of misclassified text

Methodology	Actual	Prediction
prayam oru number maathram thalivaa	Malayalam Code Mix	Tamil Code Mix
feel good movie le best movie aavum ithu urappu	Malayalam Code Mix	Tamil Code Mix
looking like little chekka chevantha vanam	Tamil Code Mix	Malayalam Code Mix
cinema kàndathinnuu shashaammm trailer kannunnnaa arengillummm ondoo	Malayalam Code Mix	Tamil Code Mix

4 Conclusion

The process of detecting Dravidian languages, namely Tamil, Tamil code-mix, Malayalam, and Malayalam code-mix, from textual data has been implemented using different machine learning algorithms and deep learning algorithms. An empirical analysis of the results obtained using the different models has also been presented. The best performance with an accuracy of 98.6% has been achieved using language agnostic based deep learning models. This accuracy is higher than the accuracy of the existing methodologies. Comparing the performances of different ML and transformer models, it could be concluded that the transformer model provides better results for the custom data set used. For language detection in code mixed Dravidian language cross lingual model has outperformed mBERT which is a multiple mono lingual model.

In the future, a gold standard dataset could be developed for the task of language detection, considering Indian languages. Also, the proposed system makes use of only four different classes of languages for classification, namely Tamil, code-mixed Tamil, Malayalam, and code-mixed Malayalam, which can be extended in the future by taking into account the other Indian languages.

References

1) Harish BS, Rangan RK. A comprehensive survey on Indian regional language processing. *SN Applied Sciences*. 2020;2(7):1–16. Available from: https://doi.org/10.1007/s42452-020-2983-x. doi:10.1007/s42452-020-2983-x.
2) Shanmugavadivel K, Sathishkumar VE, Raja S, Lingaiah TB, Neelakandan S, Subramanian M. Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. *Scientific Reports*. 2022;12(1):1–12. Available from: https://doi.org/10.1038/s41598-022-26092-3. doi:10.1038/s41598-022-26092-3.
3) Subramanian M, Sathiskumar VE, Deepalakshmi G, Cho J, Manikandan G. A survey on hate speech detection and sentiment analysis using machine learning and deep learning models. *Alexandria Engineering Journal*. 2023;80:110–121. Available from: https://doi.org/10.1016/j.aej.2023.08.038. doi:10.1016/j.aej.2023.08.038.
4) Shanmugavadivel K, Sathishkumar VE, Raja S, Lingaiah TB, Neelakandan S, Subramanian M. Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. *Scientific Reports*. 2022;12(1):1–12. Available from: https://doi.org/10.1038/s41598-022-26092-3. doi:10.1038/s41598-022-26092-3.
5) Anbukkarasi S, Elangovan D, Periyasamy J, Sathishkumar VE, Dharinya SS, Kumar MS, et al. Phonetic-Based Forward Online Transliteration Tool from English to Tamil Language. *International Journal of Reliability, Quality and Safety Engineering*. 2023;30(03). Available from: https://dx.doi.org/10.1142/s021853932350002x. doi:10.1142/s021853932350002x.
6) Anbukkarasi S, Sathishkumar VE, Dhivyaa CR, Cho J. Enhanced Feature Model Based Hybrid Neural Network for Text Detection on Signboard, Billboard and News Tickers. *IEEE Access*. 2023;11:41524–41534. Available from: https://dx.doi.org/10.1109/access.2023.3264569. doi:10.1109/access.2023.3264569.
7) Tagg C. English language and social media. In: The Routledge Handbook of English Language and Digital Humanities. Routledge. 2020;p. 568–586. Available from: https://www.taylorfrancis.com/chapters/edit/10.4324/9781003031758-30/english-language-social-media-caroline-tagg.
8) Joshi R, Joshi R. Evaluating Input Representation for Language Identification in Hindi-English Code Mixed Text. In: ICDSMLA 2020;vol. 783 of Lecture Notes in Electrical Engineering. Singapore. Springer. 2021;p. 795–802. Available from: https://doi.org/10.1007/978-981-16-3690-5_73.
9) Goyal V, Rani S, Neetika. Automatic understanding of code mixed social media text: A state of the art. In: Advances in Information Communication Technology and Computing;vol. 135 of Lecture Notes in Networks and Systems. 2020;p. 91–100. Available from: https://doi.org/10.1007/978-981-15-5421-6_10.

10) Aguilar G, Kar S, Solorio T. Lince: A centralized benchmark for linguistic code-switching evaluation. 2020. Available from: https://doi.org/10.48550/arXiv.2005.04322.

11) Chakravarthi BR, Jose N, Suryawanshi S, Sherly E, McCrae JP. A Sentiment Analysis Dataset for Code-Mixed Malayalam-English. 2020. Available from: https://doi.org/10.48550/arXiv.2006.00210.

12) Jauhiainen T, Ranasinghe T, Zampieri M. Comparing Approaches to Dravidian Language Identification. In: Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects. 2021;p. 120–127. Available from: https://aclanthology.org/2021.vardial-1.14.pdf.

13) Ansari MZ, Beg MMS, Ahmad T, Khan MJ, and GW. Language identification of hindi-english tweets using code-mixed bert. In: 2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE. 2022;p. 248–252. Available from: https://doi.org/10.1109/ICCICC53683.2021.9811292. doi:10.1109/ICCICC53683.2021.9811292.

14) Ceolin A. Neural networks for cross-domain language identification. phlyers@ vardial 2022. In: Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects. Association for Computational Linguistics. 2022;p. 99–108. Available from: https://aclanthology.org/2022.vardial-1.11.

15) Bestgen Y. Optimizing a supervised classifier for a difficult language identification problem. In: Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects. 2021;p. 96–101. Available from: https://aclanthology.org/2021.vardial-1.11.pdf.

16) Chakravarthi BR, Priyadharshini R, Muralidaran V, Jose N, Suryawanshi S, Sherly E, et al. DravidianCodeMix: sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text. *Language Resources and Evaluation*. 2022;56(3):765–806. Available from: https://doi.org/10.1007/s10579-022-09583-7. doi:10.1007/s10579-022-09583-7.

17) Sundar A, Ramakrishnan A, Balaji A, Durairaj T. Hope Speech Detection for Dravidian Languages Using Cross-Lingual Embeddings with Stacked Encoder Architecture. *SN Computer Science*. 2022;3(1):1–15. Available from: https://dx.doi.org/10.1007/s42979-021-00943-8. doi:10.1007/s42979-021-00943-8.

18) Kedia K, Nandy A. indicnlp@kgp at DravidianLangTech-EACL2021: Offensive Language Identification in Dravidian Languages. In: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics. 2021;p. 330–335. Available from: https://aclanthology.org/2021.dravidianlangtech-1.48.

19) Chakravarthi BR, Mihaela G, Ionescu RT, Jauhiainen H, Jauhiainen T, Lindén K, et al. Findings of the vardial evaluation campaign 2021. In: Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects. The Association for Computational Linguistics. Association for Computational Linguistics. 2021;p. 1–11. Available from: https://aclanthology.org/2021.vardial-1.1.

20) Sarlis S, Maglogiannis I. On the Reusability of Sentiment Analysis Datasets in Applications with Dissimilar Contexts. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, AIAI 2020;vol. 583 of IFIP Advances in Information and Communication Technology. Springer, Cham. 2020;p. 409–418. Available from: https://doi.org/10.1007/978-3-030-49161-1_34.

21) Xu S. Bayesian Naive Bayes classifiers to text classification. *Journal of Information Science*. 2018;44(1):48–59. Available from: https://doi.org/10.1177/0165551516677946. doi:10.1177/0165551516677946.

22) and MKC. Introduction to Logistic Regression. 2020. Available from: https://arxiv.org/pdf/2008.13567.pdf.

23) Sarker I. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*. 2021;2(3):1–21. Available from: https://doi.org/10.1007/s42979-021-00592-x. doi:10.1007/s42979-021-00592-x.

24) Hashimoto EM, Ortega EMM, Cordeiro GM, Suzuki AK, Kattan MW. The multinomial logistic regression model for predicting the discharge status after liver transplantation: estimation and diagnostics analysis. *Journal of Applied Statistics*. 2020;47(12):2159–2177. Available from: https://dx.doi.org/10.1080/02664763.2019.1706725. doi:10.1080/02664763.2019.1706725.

25) Alcaraz J, Labbé M, Landete M. Support Vector Machine with feature selection: A multiobjective approach. *Expert Systems with Applications*. 2022;204:1–14. Available from: https://dx.doi.org/10.1016/j.eswa.2022.117485. doi:10.1016/j.eswa.2022.117485.

26) Emmanuel T, Maupong T, Mpoeleng D, Semong T, Mphago B, Tabona O. A survey on missing data in machine learning. *Journal of Big Data*. 2021;8(1):1–37. Available from: https://doi.org/10.1186/s40537-021-00516-9. doi:10.1186/s40537-021-00516-9.

27) Rajalaxmi RR, Prasad LVN, Janakiramaiah B, Pavankumar CS, Neelima N, Sathishkumar VE. Optimizing Hyperparameters and Performance Analysis of LSTM Model in Detecting Fake News on Social media. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 2022;p. 1–17. Available from: https://dx.doi.org/10.1145/3511897. doi:10.1145/3511897.

28) Xu N, Gui T, Ma R, Zhang Q, Ye J, Zhang M, et al. Cross-Linguistic Syntactic Difference in Multilingual BERT: How Good is It and How Does It Affect Transfer?. 2022. Available from: https://doi.org/10.48550/arXiv.2212.10879.

29) Feng F, Yang Y, Cer D, Arivazhagan N, Wang W. Language-agnostic BERT Sentence Embedding. 2020. Available from: https://doi.org/10.48550/arXiv.2007.01852.