ILID: Native Script Language Identification for Indian Languages

Yash Ingle, Pruthwik Mishra
Sardar Vallabhbhai National Institute of Technology, Surat, India
yash.ingle003@gmail.com,pruthwikmishra@aid.svnit.ac.in

Abstract

The language identification task is a crucial fundamental step in NLP. Often it serves as a pre-processing step for widely used NLP applications such as multilingual machine translation, information retrieval, question and answering, and text summarization. The core challenge of language identification lies in distinguishing languages in noisy, short, and code-mixed environments. This becomes even harder in case of diverse Indian languages that exhibit lexical and phonetic similarities, but have distinct differences. Many Indian languages share the same script making the task even more challenging. In this paper, we release a dataset of 230K sentences consisting of English and all 22 official Indian languages labeled with their language identifiers where data in most languages are newly created. We also develop and release robust baseline models using state-of-the-art approaches in machine learning and deep learning that can aid the research in this field. Our baseline models are comparable to the state-of-the-art models for the language identification task.

1 Introduction

India is a linguistically diverse country. Although there are more than 1000 languages in the Indian subcontinent, the digital divide among many Indian languages is enormous. Barring a few languages, most of the languages suffer from resource scarcity and a simple task such as Language Identification (LID) remains a challenging task (Caswell et al., 2020) for them. Available LID tools such as Lui and Baldwin (2011), Joulin et al. (2016a), and Team et al. (2022) perform poorly on Indian languages and they do not cover many of them. With the increase in mobile and internet users in India, the need for providing services in native Indian languages is more pressing than ever. Hence, it becomes essential to create robust LID tools to cater to the Indian users. As the volume and variety of data keeps growing, we need to include data from different domains while creating sound benchmarks. Keeping this motivation in mind, we set out to create an Indian Language Identification (ILID) benchmark including English and 22 official languages consisting of a total 230K sentences labeled with their language markers. The main contribution of our paper is two folds:

- We create ILID dataset consisting of newly created datasets in 13 languages and curated datasets in rest of the languages.
- We develop ILID baseline models exploring different machine learning and deep learning techniques.

2 ILID Dataset Creation

Indian Language Identification (ILID) dataset is created using two approaches. We include English and all 22 official Indian languages [1] widely used in India. The first approach utilizes web scraping for the languages in which the digital presence is significant (details are given in the Appendix). For each of these languages, we collect 10,000 different sentences from diverse sources such as government websites, newspapers, books, and other public materials ensuring varying degrees of linguistic complexity. The second approach samples sentences from an existing massive monolingual and parallel corpora for Indian languages (Mujadia and Sharma, 2024). The details of the dataset are presented in Table 1. The data in each language is split into 80:10:10 ratio to create train, dev, and test sets.

To ensure quality and consistency, the dataset undergoes several noise removal steps. The first step involves the elimination of duplicate, very short, and ungrammatical sentences. The next step employs an existing FastText (Joulin et al., 2016b,a)

[1] https://en.wikipedia.org/wiki/Eighth_Schedule_to_the_Constitution_of_India

Language	#Train	#Dev	#Test	#Total
Assamese (asm)	8000	1000	1000	10000
Bengali (ben)	8000	1000	1000	10000
Bodo (brx)	8000	1000	1000	10000
Dogri (doi)	8000	1000	1000	10000
Konkani (gom)	8000	1000	1000	10000
Gujarati (guj)	8000	1000	1000	10000
Hindi (hin)	8000	1000	1000	10000
Kannada (kan)	8000	1000	1000	10000
Kashmiri (kas)	8000	1000	1000	10000
Maithili (mai)	8000	1000	1000	10000
Malayalam (mal)	8000	1000	1000	10000
Marathi (mar)	8000	1000	1000	10000
Manipuri (mni)	8000	1000	1000	10000
Nepali (npi)	8000	1000	1000	10000
Odia (ory)	8000	1000	1000	10000
Punjabi (pan)	8000	1000	1000	10000
sanskrit(san)	8000	1000	1000	10000
Santali (sat)	8000	1000	1000	10000
Sindhi (snd)	8000	1000	1000	10000
Tamil (tam)	8000	1000	1000	10000
Telugu (tel)	8000	1000	1000	10000
Urdu (urd)	8000	1000	1000	10000
English (eng)	8000	1000	1000	10000
Total	**184000**	**23000**	**23000**	**230000**

Table 1: Data splits for ILID Benchmark Dataset

based language identification model to remove sentences where the probability of detecting the language of the text is very low (we fix a threshold of 0.7 for this filtering process). This happens in code-mixed texts as many Indians are multilingual [2] and often use more than one language while writing.

The first approach is described in details.

2.1 Web Scraping

To build a strong language identification system for Indian languages, we carefully built a personalized set of 10,000 text samples for each language, covering 12 Indian languages. Details are available in Appendix A Table 4.

Text data is collected from a variety of publicly available and diverse sources, including Wikipedia dumps in respective languages for formal and structured text, news websites and blogs for professional and personal updates. All efforts are made to ensure the samples are multilingual in length, script (i.e., Devanagari, Bengali, Tamil, Odia etc.), domain, and style, reflecting the multilingual nature of India.

For a target website W, we define the scraping process as:

$$S(W) = \bigcup_{p \in P_W} \phi(p) \tag{1}$$

where P_W is the set of target pages and ϕ is the extraction function that parses HTML content while preserving structural information. We implement adaptive throttling using:

$$\Delta_t = \frac{1}{|W|} \sum_{i=1}^{|W|} \frac{\text{size}(w_i)}{\text{bandwidth}} \tag{2}$$

2.2 Data Cleaning Pipeline

In the data cleaning process we mainly work on removing extra spaces, extra special symbols, unicode normalization, breaking the paragraph into

[2] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

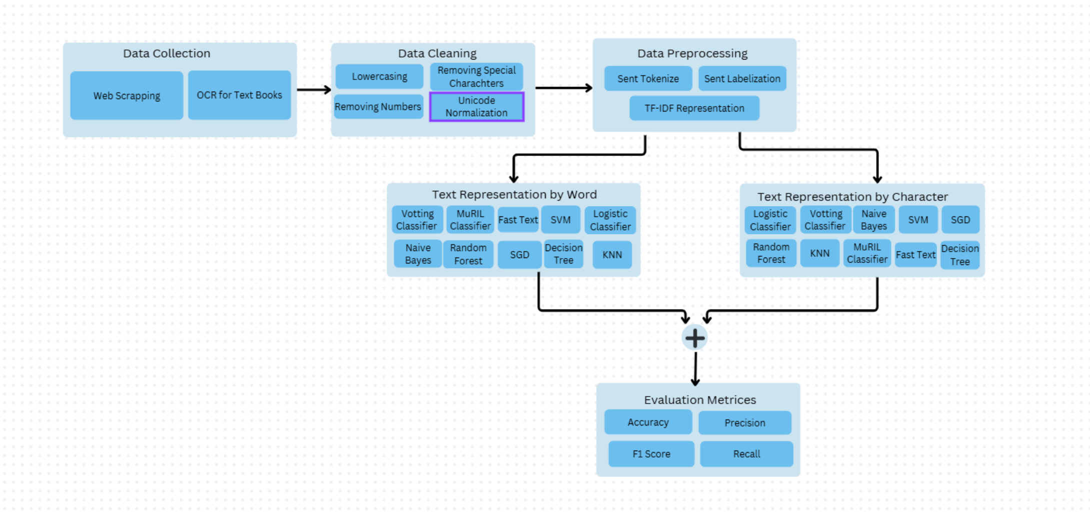

Figure 1: Workflow of the Proposed Language Identification System.

meaningful sentences, and after that we apply various tokenization methods. For sentence tokenization, we utilize the corresponding end sentence markers defined for respective languages. We use a regular expression based tokenizer [3] for sentence and word tokenization. Our cleaning process transforms raw text T to cleaned text T' through:

$$T' = \psi_n(\psi_s(T)) \tag{3}$$

where ψ_s handles special characters and ψ_n performs normalization. We use a sentence as a unit for the LID task. Each sentence is labeled with a language tag. ISO 639-2 codes [4] or 3 lettered language codes denote each language detailed in Appendix B Table 5. In the second approach, sentences are randomly sampled from the huge corpora (Mujadia and Sharma, 2024) that contain the information about the language. A similar exercise of appending the labels to the sentences as language identifiers and data splitting is performed on these samples.

3 ILID Model

The ILID model is a classifier that can categorize a piece of text into one of the 23 classes that includes English and 22 Indian languages. The architecture of the proposed system is shown in Figure 1. We implement three kinds of approaches for designing the classifiers.

3.1 Machine Learning Models

In this approach, each sentence is represented by a TF-IDF (Sparck Jones, 1972) vector in a bag-of-words setting. We utilize both word-level and character-level TF-IDF representations to enhance textual feature extraction. Word-level TF-IDF captures the importance of entire words across the corpus, which helps in preserving semantic meaning and understanding context at a higher level. However, word-level models may struggle with noisy or morphologically rich languages, spelling errors, or out-of-vocabulary terms. To address this, we also incorporate character-level TF-IDF, which analyzes smaller units such as character n-grams. This enables the model to recognize sub-word patterns and partial word structures, making it more resilient to variations and linguistic nuances. By combining both approaches, our system is able to capture both the global meaning (through words) and the finer-grained patterns (through characters), resulting in a more robust and language-agnostic representation, especially beneficial for multilingual or noisy text data. Eight different classifiers such as Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Naive Bayes', Stochastic Gradient Descent (SGD), K-Nearest Neighbors, and Adaboost are explored for this task. We develop different ensemble models consisting of 3, 4, and 5 classifiers and evaluate their performance. Due to paucity of space, we only report the five best performing ensemble models.

[3] https://github.com/Pruthwik/Tokenizer_for_Indian_Languages
[4] https://en.wikipedia.org/wiki/List_of_ISO_639-2_codes

Model	KNN	DT	RF	SVM	NB	LogReg	SGD	F1 (Dev)	F1 (Test)
Voting-1	✓	✓		✓	✓	✓		0.99	0.96
Voting-2	✓	✓	✓	✓		✓		0.99	0.96
Voting-3	✓	✓		✓		✓	✓	0.99	0.96
Voting-4	✓	✓	✓	✓	✓			0.99	0.96
Voting-5			✓	✓	✓	✓	✓	0.99	0.99
MuRIL								0.96	0.96
FastText								0.96	0.96

Table 2: Comparison of Voting Classifiers, MuRIL, and FastText on Dev and Test sets

Model	KNN	DT	RF	SVM	NB	LogReg	SGD	ILID	IndicLID
Voting-1	✓	✓		✓	✓	✓		0.98	0.94
Voting-2	✓	✓	✓	✓		✓		0.98	0.96
Voting-3	✓	✓		✓		✓	✓	0.98	0.95
Voting-4	✓	✓	✓	✓	✓			0.98	0.95
Voting-5			✓	✓	✓	✓	✓	0.98	0.97
MuRIL								0.98	0.94
FastText								0.98	0.92

Table 3: Comparison of Voting Classifiers, MuRIL, and FastText on Other Benchmarks

3.2 FastText Classifier

FastText (Joulin et al., 2016b,a) utilizes word embeddings (Bojanowski et al., 2017) composed of character n-grams that can better represent rare words and orthographic similarities. FastText classifier is a linear classifier and is very fast at computation.

4 Pretrained BERT Model

Pretrained subword based contextual language models such as BERT (Devlin et al., 2019), XLM (Conneau et al., 2020), RoBERTa (Zhuang et al., 2021) have proven to be very effective in text classification tasks or generally natural language understanding tasks. One variant of BERT, MuRIL (Khanuja et al., 2021) is pretrained on Indian languages data. Hence, we fine-tune the MuRIL pretrained model on the ILID train set for our task.

5 Experimental Details

The machine learning models have been implemented using the Scikit-learn (Pedregosa et al., 2011) framework. Similarly, the FastText library [5] is utilized to implement the classification on the languages of the texts. The MuRIL models is fine-tuned using the Hugginggace transformer (Wolf et al., 2019) framework. We will upload the models with MIT license once the paper gets accepted.

6 Evaluation Metrics

All models have been evaluated using the macro F1 scores. The macro F1 score averages the F1 scores across all the languages. In order to evaluate the score, the precision and recall scores are also computed at the macro level. Each ensemble employs majority voting mechanism to determine the predicted labels.

7 Results and Discussion

The ensemble machine learning models perform better than the individual models. The performance of the ensembles is also superior to the FastText and fine-tuned MuRIL models. The best five performing ensembles along with the FastText and fine-tuned MuRIL models are presented in Table 2

[5] https://fasttext.cc/docs/en/supervised-tutorial.html

that represent the macro F1 score for each model. The results for the individual models are shown in Tables 6 and 7. Although the scores are encouraging, the performance drops in languages that share a script such as Devanagari for Hindi, Maithili, Marathi, Konkani, and Sanskrit, Arabic for Kashmiri, Urdu, and Sindhi. We compare our models with the state-of-the-art IndicLID model (Madhani et al., 2023) on the Bhasha-Abhijnaanam (Madhani et al., 2023) benchmark where our model perform comparatively. Our ILID model perform better on the languages where data are scraped while the performance dips for languages that are created from other external resources.

8 Conclusion

In this paper, we create the ILID benchmark dataset for English and Indian languages and develop baseline language identification models. The dataset and models will serve as foundation tools mostly for the resource poor languages that are part of this benchmark. We hope that this will act as an impetus for future NLP research in these languages.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 553–561, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Yash Madhani, Mitesh M. Khapra, and Anoop Kunchukuttan. 2023. Bhasa-Abhijnaanam: Native-script and romanized language identification for 22 Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 816–826, Toronto, Canada. Association for Computational Linguistics.

Vandan Mujadia and Dipti Misra Sharma. 2024. Bhashaverse : Translation ecosystem for indian subcontinent languages.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Languages Used in Scraping and Sampling

As explained in 2, based on their online availability, we used two approaches of scraping and sampling from an existing corpus for the dataset creation. The languages are shown in Table 4.

Scraping	Sampling
Assamese	Bodo
Bengali	Dogri
Gujarati	Kashmiri
Hindi	Konkani
Kannada	Maithili
Malayalam	Manipuri
Marathi	Nepali
Oriya	Sanskrit
Punjabi	Santhali
Tamil	Sindhi
Telugu	
Urdu	
English	

Table 4: Languages Used in Scraping and Sampling

B ISO 639-2 Codes

We utilize ISO 639-2 codes as the language labels for this task.

Language	ISO 639-2 Code
Assamese	asm
Bangla	ben
Bodo	brx
Dogri	doi
Gujarati	guj
Hindi	hin
Kannada	kan
Kashmiri	kas
Konkani	gom
Maithili	mai
Malayalam	mal
Manipuri	mni
Marathi	mar
Nepali	npi
Oriya	ory
Punjabi	pan
Sanskrit	san
Santali	sat
Sindhi	snd
Tamil	tam
Telugu	tel
Urdu	urd
English	eng

Table 5: Language Full Name to ISO 639-2 Language Codes Mapping

C Results for Individual Languages

The language wise performance of the models are shown in Tables 6 and 7.

Language	**LR**	**DT**	**RF**	**SVM**	**SGD**	**k-NN**	**NB**	**ADABoost**	**MURIL**	**FastText**
Assamese (asm)	**0.99**	0.97	**0.99**	**0.99**	0.98	0.91	0.98	0.77	**0.99**	**0.99**
Bengali (ben)	**1.00**	0.99	0.99	**1.00**	0.99	0.93	0.99	0.92	**1.00**	**1.00**
Bodo (brx)	0.96	0.92	0.96	**0.97**	0.94	0.90	0.96	0.68	**0.97**	0.96
Dogri (doi)	**0.90**	0.82	0.89	**0.90**	0.88	0.71	**0.90**	0.19	**0.95**	0.92
English (eng)	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.99	0.95	**1.00**	**1.00**
Konkani (gom)	**0.97**	0.85	0.95	**0.97**	0.93	0.80	0.96	0.29	**0.98**	**0.98**
Gujarati (guj)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
Hindi (hin)	**0.99**	0.93	**0.99**	**0.99**	0.98	0.83	0.94	0.79	**1.00**	0.99
Kannada (kan)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
Kashmiri (kas)	**0.99**	0.95	0.97	**0.99**	0.94	0.97	0.98	0.75	**1.00**	**1.00**
Maithili (mai)	**0.87**	0.78	0.84	0.86	0.84	0.67	0.84	0.26	**0.92**	0.90
Malayalam (mal)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
Marathi (mar)	**0.98**	0.84	0.96	**0.98**	0.93	0.76	0.94	0.52	**0.99**	**0.99**
Manipuri (mni)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	0.72	0.72
Nepali (npi)	**0.92**	0.84	0.90	0.91	0.86	0.76	0.89	0.28	**0.94**	0.92
Odia (ory)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
Punjabi (pan)	**1.00**	0.99	**1.00**	**1.00**	0.99	0.99	0.99	0.91	**1.00**	**1.00**
Sanskrit (san)	**0.97**	0.92	**0.99**	**0.98**	0.97	0.84	0.94	0.78	**0.99**	0.99
Santali (sat)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.71	0.71
Sindhi (snd)	**0.94**	0.86	0.92	0.93	0.90	0.80	0.94	0.24	**0.96**	0.94
Tamil (tam)	**1.00**	0.99	**1.00**	**1.00**	0.99	0.99	0.99	0.95	**1.00**	**1.00**
Telugu (tel)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.91	**1.00**	**1.00**
Urdu (urd)	**0.99**	0.94	0.96	**0.99**	0.96	0.97	0.98	0.76	**1.00**	**1.00**

Table 6: Dev dataset report: word+char combined Classifier performance (F1-scores) across 22 languages. Bold indicates top-performing model(s) per language.

Language	LR	DT	RF	SVM	SGD	k-NN	NB	AdaBoost	MuRIL	FastText
Assamese (asm)	0.99	0.96	0.98	**0.99**	0.98	0.91	0.97	0.78	**0.99**	**0.99**
Bengali (ben)	**1.00**	0.98	0.99	**1.00**	0.99	0.93	0.99	0.93	**1.00**	**1.00**
Bodo (brx)	0.95	0.90	**0.96**	**0.96**	0.95	0.89	0.96	0.69	0.97	0.96
Dogri (doi)	**0.91**	0.82	**0.91**	0.90	0.88	0.71	**0.91**	0.21	0.94	0.92
English (eng)	**1.00**	**1.00**	0.99	**1.00**	0.99	**1.00**	0.99	0.96	**1.00**	**1.00**
Konkani (gom)	**0.97**	0.86	0.94	**0.97**	0.92	0.81	0.95	0.31	0.98	0.97
Gujarati (guj)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
Hindi (hin)	0.99	0.93	0.98	**0.99**	0.97	0.84	0.94	0.74	0.99	0.99
Kannada (kan)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**1.00**	**1.00**
Kashmiri (kas)	0.98	0.93	0.97	0.98	0.94	**0.98**	**0.98**	0.74	**1.00**	0.99
Maithili (mai)	**0.85**	0.78	0.84	**0.85**	0.81	0.67	0.82	0.26	0.91	0.89
Malayalam (mal)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**
Marathi (mar)	0.97	0.85	0.95	**0.98**	0.92	0.79	0.94	0.50	0.98	0.98
Manipuri (mni)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.89	0.73	0.72
Nepali (npi)	0.90	0.84	0.90	0.90	0.86	0.76	**0.89**	0.28	**0**.94	0.91
Odia (ory)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
Punjabi (pan)	**1.00**	0.99	0.99	**1.00**	0.99	0.99	**1.00**	0.91	**1.00**	**1.00**
Sanskrit (san)	**0.98**	0.92	**0.99**	**0.98**	0.97	0.84	0.94	0.79	**0.98**	0.98
Santali (sat)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.94	0.72	0.71
Sindhi (snd)	0.94	0.87	0.93	0.93	0.90	0.80	**0.92**	0.23	0.96	0.94
Tamil (tam)	**1.00**	0.99	**1.00**	**1.00**	0.99	**1.00**	**1.00**	0.96	**1.00**	**1.00**
Telugu (tel)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.92	**1.00**	**1.00**
Urdu (urd)	0.98	0.92	0.97	**0.99**	0.96	0.98	**0.98**	0.75	**1.00**	**0.99**

Table 7: Test dataset report showing word+char combined classifier performance (F1-scores) across 22 languages. Bold indicates top-performing model(s) per language.