Bhasha-Abhijnaanam: Native-script and Romanized Language Identification for 22 Indic Languages

Yash Madhani[1] **Mitesh M. Khapra**[2] **Anoop Kunchukuttan**[3]
AI4Bharat[1,2,3] IIT Madras[1,2,3] Microsoft[3]
[1]cs20s002@cse.iitm.ac.in [2]miteshk@cse.iitm.ac.in [3]ankunchu@microsoft.com

Abstract

We create publicly available language identification (LID) datasets and models in all 22 Indian languages listed in the Indian constitution in both native-script and romanized text. First, we create *Bhasha-Abhijnaanam*, a language identification test set for native-script as well as romanized text which spans all 22 Indic languages. We also train *IndicLID*, a language identifier for all the above-mentioned languages in both native and romanized script. For native-script text, it has better language coverage than existing LIDs and is competitive or better than other LIDs. IndicLID is the first LID for romanized text in Indian languages. Two major challenges for romanized text LID are the lack of training data and low-LID performance when languages are similar. We provide simple and effective solutions to these problems. In general, there has been limited work on romanized text in any language, and our findings are relevant to other languages that need romanized language identification. Our models are publicly available at `https://github.com/AI4Bharat/IndicLID` under open-source licenses. Our training and test sets are also publicly available at `https://huggingface.co/datasets/ai4bharat/Bhasha-Abhijnaanam` under open-source licenses.

1 Introduction

In this work, we focus on building a language identifier for the 22 languages listed in the Indian constitution. With increasing digitization, there is a push to make NLP technologies like translation, ASR, conversational technologies, etc. (Bose, 2022) available as a public good at population scale (Chandorkar, 2022). A good language identifier is required to help build corpora in low-resource languages. For such languages, language identification is far from a solved problem due to noisy web crawls, small existing datasets, and similarity to high-resource languages (Caswell et al., 2020).

Existing publicly available LID tools like CLD3[1], LangID[2] (Lui and Baldwin, 2011), FastText[3] (Joulin et al., 2016) and NLLB[4] (NLLB Team et al., 2022) have some shortcomings with respect to Indian languages. They do not cover all the above-mentioned 22 languages. In social media and chats, it is also common to use the roman script for most Indian languages leading to substantial user-generated content in roman script. However, none of the LIDs have any support for the detection of romanized Indian language text (except `cld3` support for Latin Hindi). The widespread use of romanization implies that accurate romanized Language Identification models are a critical component in the NLP stack for Indian languages, given that this affects over 735 million internet users (KPMG and Google, 2017). Therefore, our work on developing accurate and effective romanized Language Identification models has the potential to make a significant impact in the NLP space for Indian languages, particularly in the social media and chat application domains. Hence, we undertake the task of creating a LID for these 22 Indian languages. The main contributions of our work are as follows:

- We create *Bhasha-Abhijnaanam*[5], a language identification test set for native-script as well as romanized text which spans 22 Indic languages. Previous benchmarks for native script do not cover all these languages (NLLB Team et al., 2022; Roark et al., 2020). The Dakshina test set for romanized text covers only 11 languages and there are ambiguous instances in the test set like named entities that cannot be assigned to a particular language (Roark et al., 2020).
- We also train, *IndicLID*, an LID for all the above-

[1] https://github.com/google/cld3

[2] https://github.com/saffsd/langid.py

[3] https://fasttext.cc/docs/en/language-identification.html

[4] https://github.com/facebookresearch/fairseq/tree/nllb#lid-model

[5] The word means language-identification in Sanskrit.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 816–826
July 9-14, 2023 ©2023 Association for Computational Linguistics

mentioned languages in both native and romanized script. For native-script training data, we sample sentences from diverse sources and oversample low-resource languages. IndicLID native-script model has better language coverage than existing LIDs and is competitive or better than other LIDs with 98% accuracy and at least 6 times better throughput.

• To the best of our knowledge, ours is one of the first large-scale efforts for romanized LID in any language, a task that has not received much attention. A major challenge for romanized text LID is the lack of romanized training data. We show that synthetic romanized training data created via transliteration can help train a reasonably good LID for romanized text. A simple linear classifier does not perform well for romanized text. Hence, we combine a simple but fast text classifier with a slower but more accurate classifier based on a pre-trained language model to achieve a good trade-off between accuracy and speed.

Our findings are relevant to other languages that need LID for romanized text. We require native script data and a transliteration model to create the synthetic romanized data for the target language. This romanized data serves as training data for the romanized LID.

2 Bhasha-Abhijnaanam benchmark

We describe the creation of the Bhasha-Abhijnaanam LID benchmark for 22 Indian languages in native and roman script. Table 1 describes the statistics of the *Bhasha-Abhijnaanam* benchmark. We build upon existing benchmarks to fill in the coverage and quality gaps and cost-efficiently cover all languages.

2.1 Native script test set.

We compile a native script test set comprising 19 Indian languages and 11 scripts from the FLORES-200 devtest (NLLB Team et al., 2022) and Dakshina sentence test set (Roark et al., 2020). We create native text test sets for the remaining three languages (*Bodo, Konkani, Dogri*) and one script (*Manipuri* in *Meetei Mayek* script) not covered in these datasets. For these new languages we first sample the English sentences from Wikipedia and ask in-house, professional translators to translate the sentences to respective languages. This method ensured the quality and accuracy of our test samples, as well as minimizing any potential noise in the data.

Language	**Script**	**Native**	**Roman**
Assamese	Bengali	1012	**512**
Bangla	Bengali	5606	4595
Bodo	Devanagari	**1500**	**433**
Dogri	Devanagari	**1498**	**512**
Gujarati	Gujarati	5797	4785
Hindi	Devanagari	5617	4606
Kannada	Kannada	5859	4848
Kashmiri	Perso-Arabic	2511	**450**
	Devanagari	1012	
Konkani	Devanagari	**1500**	**444**
Maithili	Devanagari	2512	**439**
Malayalam	Malayalam	5628	4617
Manipuri	Bengali	1012	**442**
	Meetei Mayek	**1500**	
Marathi	Devanagari	5611	4603
Nepali	Devanagari	2512	**423**
Oriya	Oriya	1012	**512**
Punjabi	Gurmukhi	5776	4765
Sanskrit	Devanagari	2510	**448**
Santali	Ol Chiki	2512	0
Sindhi	Perso-Arabic	5893	4881
Tamil	Tamil	5779	4767
Telugu	Telugu	5751	4741
Urdu	Perso-Arabic	6883	4371

Table 1: Summary of the Bhasha-Abhijnaanam benchmark. Number of romanized and native-script sentences are reported. The cells in **bold** indicate the datasets newly contributed by this work. Romanized Santali testset has not been created since Santhali annotators we contacted did not use roman script and spoke Bengali as a second language. NLLB Team et al. (2022) also cite a similar experience.

2.2 Roman script test set.

We propose a new benchmark test set to evaluate roman-script language identification for 21 Indian languages. Out of these, 11 languages are represented in the Dakshina romanized sentence test set (Roark et al., 2020), which comprises native script sentences from Wikipedia along with their romanization. However, this test set includes short sentences which are just named entities and English loan words which are not useful for romanized text LID evaluation. To address this issue, we manually validate the Dakshina test sets for the languages we are interested in and filter out about 7% of the sentences. Section 2.3 describes the details of the filtering process. To create a benchmark test set for the remaining 10 Indian languages, we sampled sentences from IndicCorp (Doddapaneni et al.,

Language	Total samples	Valid samples	%filtered
Bengali	5001	4600	8.0183
Gujarati	5001	4789	4.2391
Hindi	5001	4616	7.6984
Kannada	5001	4849	3.0393
Malayalam	5001	4627	7.4785
Marathi	5001	4617	7.6784
Punjabi	5001	4782	4.3791
Sindhi	5001	4889	2.2395
Tamil	5001	4802	3.9792
Telugu	5001	4754	4.9390
Urdu	4881	4395	9.9569

Table 2: Statistics of Dakshina roman filtered test set

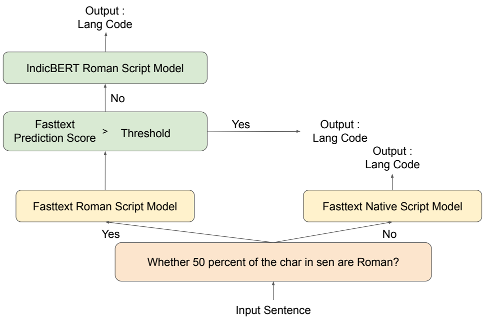
Output :
Lang Code
IndicBERT Roman Script Model
No
Fasttext
Prediction Score
>
Threshold
Yes
Output :
Lang Code
Output :
Lang Code
Fasttext Roman Script Model
Fasttext Native Script Model
Yes
No
Whether 50 percent of the char in sen are Roman?
Input Sentence

Figure 1: IndicLID Classifier Workflow

2022) and asked annotators to write the same in roman script. We did not specify any transliteration guidelines and annotators were free to transliterate in the most natural way they deemed fit. We additionally asked annotators to skip the sentence if they find it invalid (wrong language, offensive, truncated, etc.).

2.3 Romanized Dakshina testset filtering

The Dakshina romanized sentence test set includes short sentences which are just named entities and English loan words which are not useful for romanized text LID evaluation. To address this issue, we manually validate the Dakshina test sets for the languages we are interested in. We first identified potentially problematic sentences from the romanized Dakshina test set by applying two constraints: (i) sentences shorter than 5 words, and (ii) native LID model is less confident about the native language sentence (prediction score less than 0.8). These sentences were then validated by native language annotators. The annotators were asked to read the roman sentences and determine whether they were named entities or sentences where they could not determine the language. Such entries were filtered out. About 7% of the sentences were filtered. Table 2 describes the filtering statistics.

3 IndicLID Model

IndicLID is a classifier specifically for Indic languages that can predict 47 classes (24 native-script classes and 21 roman-script classes plus English and Others). We create three classifier variants: a fast linear classifier, a slower classifier finetuned from a pre-trained LM, and an ensemble of the two models which trades off speed v/s accuracy.

3.1 Training dataset creation

Native-script training data. We compiled the training data sentences from various sources viz. IndicCorp (Doddapaneni et al., 2022), NLLB (NLLB Team et al., 2022), Wikipedia, Vikaspedia [6] and internal sources. To ensure a diverse and representative training dataset, we sampled 100k sentences per language-script combination in a balanced way across all these sources. We used oversampling for languages with less than 100k sentences. We tokenized and normalized the sentences using IndicNLP library [7] (Kunchukuttan, 2020) with default settings.

Romanized training data. There is hardly any romanized corpora for Indian languages in the public domain[8]. Hence, we explored the use of transliteration for creating synthetic romanized data. We create romanized training data by transliterating the native script training data into roman script using the multilingual IndicXlit[9] transliteration model (Indic-to-En version) (Madhani et al., 2022), The authors have provided results on the transliteration quality of the IndicXlit model. We rely on this analysis to ensure the quality of generated training data.

3.2 Linear classifier

Linear classifiers using character n-gram features are widely used for LIDs (Jauhiainen et al., 2021). We use FastText (Joulin et al., 2016) to train our fast, linear classifier. It is a lightweight and efficient linear classifier that is well-suited for handling large-scale text data. It utilizes character n-gram features which enables it to utilize subword information. This makes it particularly useful for dealing with rare words and allows it to discriminate between similar languages having sim-

[6] https://vikaspedia.in

[7] https://github.com/anoopkunchukuttan/indic_nlp_library

[8] CC-100 has romanized versions for 4 Indian languages, but a manual analysis suggested that it contains a lot of profane content.

[9] https://github.com/AI4Bharat/IndicXlit

ilar spellings. We trained separate classifiers for native script (**IndicLID-FTN**) and roman script (**IndicLID-FTR**). We chose 8-dimension word-vector models after experimentation as they maintain small model sizes without losing model accuracy (refer Appendix A for results).

3.3 Pretrained LM-based classifier

For romanized text, we observed that linear classifiers do not perform very well. Hence, we also experimented with models having larger capacity. Particularly, we finetuned a pretrained LM on the romanized training dataset. We evaluated the following LMs: XLM-R (Conneau et al., 2020), IndicBERT-v2 (Doddapaneni et al., 2022) and MuRIL (Khanuja et al., 2021). The last two LMs are specifically trained for Indian languages and MuRIL also incorporates synthetic romanized data in pre-training. Hyperparameters for finetuning are described in Appendix B. We used IndicBERT-based classifier as the LM-based classifier (henceforth referred to as **IndicLID-BERT**) since it was amongst the best-performing romanized text classifiers and had maximum language coverage.

3.4 Final Ensemble classifier

Our final IndicLID classifier is an pipeline of multiple classifiers. Figure 1 shows the overall workflow of the IndicLID classifier. The pipeline works as described here: (1) Depending on the amount of roman script in the input text, we invoke either the native-text or romanized linear classifier. IndicLID-FTR is invoked for text containing >50% roman characters. (2) For roman text, if IndicLID-FTR is not confident about its prediction, we redirect the request to the IndicLID-BERT. We resort to this two-stage approach for romanized input to achieve a good trade-off between classifier accuracy and inference speed. The fast IndicLID-FTR's prediction is used if the model is confident about its prediction (probability of predicted class $>$ 0.6), else the slower but more accurate IndicLID-BERT is invoked. This threshold provides a good trade-off (See Appendix C for more details).

4 Results and Discussion

We discuss the performance of various models on the benchmark and analyze the results. To prevent any overlap between the test/valid and train sets, we excluded the Flores-200 test set (NLLB Team et al., 2022), Dakshina test set (Roark et al., 2020) while sampling native train samples from various sources. Additionally, we removed the training samples from the benchmark samples when collecting sentences for the benchmark test set. We also made sure that there was no overlap between the test and valid sets. To create the romanized training set, we simply transliterated the native training set. As the Dakshina test set (Roark et al., 2020) provided parallel sentences for the native and roman test sets, there was no overlap between the roman train and test sets.

Model	**P**	**R**	**F1**	**Acc**	**Throughput**	**Size**
IndicLID-FTN-8-dim (24)	98.11	98.56	98.31	98.55	30,303	318M
Comparing our IndicLID-FTN model with CLD3 model (12)						
IndicLID-FTN-4-dim	99.43	98.40	98.89	98.33	47,619	208M
IndicLID-FTN-8-dim	99.73	98.67	99.18	98.62	33,333	318M
CLD3	98.52	98.14	98.31	98.03	4,861	-
Comparing our IndicLID-FTN model with NLLB model (20)						
IndicLID-FTN-4-dim	97.78	98.10	97.92	98.19	41,666	208M
IndicLID-FTN-8-dim	98.13	98.59	98.34	98.56	29,411	318M
NLLB	99.28	98.65	98.95	98.78	4,970	1.1G

Table 3: Benchmarking on the Bhasha-Abhijnaanam native-script testset. For fair comparison with NLLB and CLD3, we restrict the comparison to languages that are common with IndicLID-FTN (count of common languages is indicated in brackets). Throughput is number of sentence/second.

4.1 Native script LID

We compare IndicLID-FTN with the NLLB model (NLLB Team et al., 2022) and the CLD3 model. As we can see in Table 3, the LID performance of IndicLID-FTN is comparable or better than other models. Our model is 10 times faster and 4 times smaller than the NLLB model. The model's footprint can be further reduced by model quantization (Joulin et al., 2016) which we leave for future work.

4.2 Roman script LID

Table 4 presents the results of different model variants on the romanized test set (see Appendix D for language-wise results). IndicLID-BERT is significantly better than IndicLID-FTR, but the throughput decreases significantly. The ensemble model (IndicLID) maintains the same LID performance as IndicLID-BERT with a 3x increase in the throughput over IndicLID-BERT. Further speedups in the model throughput can be achieved by creating distilled versions, which we leave for future work.

LID confusion analysis The confusion matrix for IndicLID is shown in Figure 2. We see that major confusions are between similar languages. Some

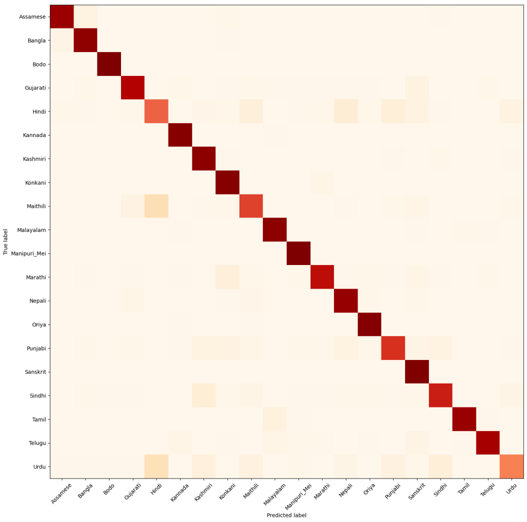

Figure 2: Confusion matrix (IndicLID, roman testset)

Model	P	R	F1	Acc	Throughput	Size
IndicLID-FTR (dim-8)	63.12	78.01	63.28	71.49	37,037	357 M
IndicLID-BERT (unfeeze-layer-1)	72.70	84.01	74.52	80.04	3	1.1 GB
IndicLID (threshold-0.6)	72.74	84.50	74.72	80.40	10	1.4 GB

Table 4: Performance of IndicLID-FTR on Bhasha-Abhijnaanam roman script test set. Throughput is number of sentence/second.

examples of such language clusters that can be observed are (1) Hindi and very close languages like Maithili, Urdu and Punjabi, (2) Konkani and Marathi, (3) Sindi and Kashmiri. Improving romanized LID between very similar languages is thus an important direction of improvement.

Impact of synthetic training data To understand the impact of synthetic training data, we generate a machine-transliterated version of the romanized test set using IndicXlit. We compare the LID accuracy on the original and synthetically generated test sets. Table 5 shows that the results on the synthetic test set are significantly better than the original test set (approaching accuracy levels in the 90s). The data characteristics of the synthetic test set are much closer to the training data than the original test set. Closing the training-test distribution gap (by representing original romanized data in the training data and/or improved generation of synthetic romanized data to reflect true data distribution) is critical to improving model performance.

Testset	P	R	F1	Acc
Original	72.74	84.50	74.72	80.40
Synthetic	90.79	97.24	93.43	95.96

Table 5: Comparison of results on Synthetic vs. original Romanized test sets for IndicLID model

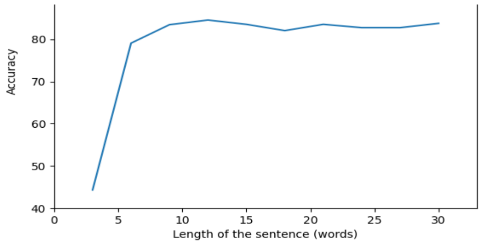

Figure 3: Effect of input length on romanized testset

The confusion matrix gives further insights into the impact of synthetic training data. Hindi is confused with languages like Nepali, Sanskrit, Marathi and Konkani using the same native script as Hindi (Devanagari). Since a multilingual transliteration model with significant Hindi data was used to create the synthetic romanized training data, it may result in the synthetic romanized forms of these languages being more similar to Hindi than would be the case with original romanized data.

Impact of input length Figure 3 plots the LID accuracy for various input length buckets. The LID is most confused for short inputs (<10 words) after which the performance is relatively stable.

5 Conclusion

We introduce an LID benchmark and models for native-script and romanized text in 22 Indian languages. These tools will serve as a basis for building NLP resources for Indian languages, particularly extremely low-resource ones that are "left-behind" in the NLP world today (Joshi et al., 2020). Our work takes first steps towards LID of romanized text, and our analysis reveals directions for future work.

Acknowledgements

We would like to thank the Ministry of Electronics and Information Technology of the Government of India for their generous grant through the Digital India Bhashini project. We also thank the Centre for Development of Advanced Computing for providing compute time on the Param Siddhi Supercomputer. We also thank Nilekani Philanthropies for their generous grant towards building datasets, models, tools and resources for Indic languages. We also thank Microsoft for their grant to support research on Indic languages. We would like to thank Jay Gala and Ishvinder Sethi for their help in coordinating the annotation work. Most importantly we would like to thank all the annotators who helped create the Bhasha-Abhijnaanam benchmark.

Limitations

The benchmark for language identification for the most part contains clean sentences (grammatically correct, single script, etc.). Data from the real world might be noisy (ungrammatical, mixed scripts, code-mixed, invalid characters, etc.). A better representative benchmark might be useful for such use cases. However, the use cases captured by this benchmark should suffice for the collection of clean monolingual corpora. This also represents a first step for many languages where no LID benchmark exists.

The use of synthetic training data seems to create a gap in performance due to divergence in train/test data distributions. Acquisition of original native romanized text and methods to generate better romanized text are needed.

Note that the romanized LID model does not support Dogri since the IndicXlit transliteration model does not support Dogri. However, since Dogri is written in the Devanagari script using the transliterator for Hindi which uses the same script might be a good approximation to generate synthetic training data. We will explore this in the future.

This work is limited to the 22 languages listed in the 8^{th} schedule of the Indian constitution. Further work is needed to extend the benchmark to many more widely used languages in India (which has about 30 languages with more than a million speakers).

Ethics Statement

For the human annotations on the dataset, the language experts are native speakers of the languages and from the Indian subcontinent. They were paid a competitive monthly salary to help with the task. The salary was determined based on the skill set and experience of the expert and adhered to the norms of the government of our country. The dataset has no harmful content. The annotators were made aware of the fact that the annotations would be released publicly and the annotations contain no private information. The proposed benchmark builds upon existing datasets. These datasets and related works have been cited.

The annotations are collected on a publicly available dataset and will be released publicly for future use. The IndicCorp dataset which we annotated has already been checked for offensive content.

All the datasets created as part of this work will be released under a CC-0 license[10] and all the code and models will be released under an MIT license.[11]

References

Arghanshu Bose. 2022. Explained: What is Bhashini and how it can bridge the gap between Indian languages. In The Times of India, 2 Sep 2022.

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.

Aashish Chandorkar. 2022. UPI, CoWIN, ONDC: Public Digital Infrastructure Has Put India on the Fast Lane of Tech-led Growth. In News18, 28 May 2022.

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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Towards Leaving No Indic Language Behind: Building Monolingual Corpora, Benchmark and Models for Indic Languages. *arXiv preprint arXiv:2212.05409.*

Tommi Jauhiainen, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Comparing approaches to dravidian language identification. *arXiv preprint arXiv:2103.05552.*

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

[10] https://creativecommons.org/publicdomain/zero/1.0

[11] https://opensource.org/licenses/MIT

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

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*.

KPMG and Google. 2017. Indian Languages - Defining India's Internet. https://assets.kpmg/content/dam/kpmg/in/pdf/2017/04/Indian-languages-Defining-Indias-Internet.pdf.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

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, Sushane Parthan, Priyanka Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *arXiv preprint arXiv:2205.03018*.

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. *arXiv preprint arXiv:2207.04672*.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith B. Hall. 2020. Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 2413–2423. European Language Resources Association.

Dimension	Precision	Recall	F1-Score	Accuracy	Throughput	Model Size
4	60.01	74.56	61.09	67.52	50000	171M
8	63.13	78.02	63.29	71.49	37037	357M
16	63.67	78.33	64.32	71.58	30303	578M
32	64.62	78.67	65.16	71.95	15625	1.6G
64	64.54	78.58	65.10	71.93	14085	1.9G
128	64.55	78.45	65.03	71.77	9901	3.3G
256	64.60	78.54	65.13	71.89	7463	7.3G
512	63.89	78.29	64.58	71.49	4608	11G
768	64.37	78.63	65.07	72.04	3876	22G
1024	64.30	78.53	65.07	71.94	3322	29G

Table 6: IndicLID-FTR performance on Bhasha-Abhijnaanam roman script test set. IndicLID-FTR are hyper-tuned by fixing different dimensions. Throughput is number of sentence/second.

Model	Precision	Recall	F1-Score	Accuracy
XLMR (Conneau et al., 2020)	63.19	70.92	59.49	65.15
MuRIL (Khanuja et al., 2021)	66.70	79.08	67.77	73.70
IndicBERT (Doddapaneni et al., 2022)	68.07	80.52	68.91	75.81

Table 7: Bhasha-Abhijnaanam roman script test set results on roman script Language models finetuned by freezing all the layers

A Hyperparameter tuning for Roman script linear classifier

We train the IndicLID-FTR model using 100k samples. While deciding the configuration IndicLID-FTR model, we experimented with fixing the dimension of IndicLID-FTR model and tuning on the rest of the hyperparameters. As we can see from table 6 model size increases with the increase of IndicLID-FTR dimension. However, beyond 8 dimensions, there is not much improvement observed. Therefore, we chose the model with 8 dimensions, taking into account the model size.

B Model selection for Roman script LM-based classifier

We experimented with three different pre-trained language models: IndicBERT (Doddapaneni et al., 2022), XLM-R (Conneau et al., 2020), and MuRIL (Khanuja et al., 2021). In the initial experiment, we froze all the layers except for the last softmax layer and finetuned the model with our training

Model	Precision	Recall	F1-Score	Accuracy
unfreezed-layer-1	72.70	84.01	74.53	80.04
unfreezed-layer-2	69.84	83.84	72.44	79.55
unfreezed-layer-4	69.53	83.44	72.12	79.47
unfreezed-layer-6	68.41	81.89	70.02	77.08
unfreezed-layer-8	67.46	81.88	68.42	76.04
unfreezed-layer-11	70.55	83.73	72.63	79.88

Table 8: Bhasha-Abhijnaanam roman script test set results on IndicLID-BERT finetuned with unfreezing different numbers of layers

Thresholds	P	R	F1	Acc	Throughput
threshold 0.1	63.13	78.02	63.29	71.49	50000
threshold 0.2	63.43	78.18	63.63	71.77	379
threshold 0.3	65.50	79.64	66.15	73.84	54
threshold 0.4	68.39	81.84	69.77	76.84	22
threshold 0.5	70.99	83.60	72.87	79.15	14
threshold 0.6	72.74	84.51	74.72	80.4	10
threshold 0.7	73.60	84.80	75.54	80.93	9
threshold 0.8	73.88	84.81	75.77	80.96	8
threshold 0.9	73.51	84.50	75.35	80.62	6

Table 9: Trade-off between inference time and accuracy with different thresholds. Throughput is number of sentence/second.

data. To fine-tune the language model, we added one softmax layer to the end of the model and used our roman script training data to finetune the model. The results for these experiments are shown in Table 7. We found that IndicBERT and MuRIL performed similarly among these three models for our roman LID task. MuRIL leverages the advantage of roman text training data, while IndicBERT was trained on the only native script but performed similarly. However, IndicBERT supports 24 Indian languages, while MuRIL only supports 17 Indian languages. Therefore, we selected IndicBERT due to its superior coverage and performance.

We then further experimented with IndicBERT by unfreezing 1, 2, 4, 6, 8, and 11 layers. The results and comparison of all the experiments are described in Table 8. We found that unfreezing 1 layer was enough for our task and that unfreezing more layers did not provide any additional benefit.

C Analysis of speed/accuracy tradeoff

We experimented IndicLID with different thresholds. If the probability score is below a certain threshold we invoke a more powerful model IndicLID-BERT, otherwise, we go with IndicLID-FTR model prediction. IndicLID-FTR model is quite fast as compared to IndicLID-BERT model. We can see a good trade-off between throughput and accuracy in table 9 as we increase the threshold. As the threshold increases, the input is more likely to go towards the IndicLID-BERT model, as we are making the model less reliant on the IndicLID-FTR model.

D Language-wise analysis for Roman script classifiers

Table 10 illustrates the language-specific performance of IndicLID-FTR, IndicLID-BERT and IndicLID models in detail. As we can see IndicLID-BERT has better representation than IndicLID-FTR for almost all the languages which leads better F1 score for IndicLID. However, for the languages of Sanskrit and Manipuri, the IndicLID-FTR model has a better representation than the IndicLID-BERT model, which is an interesting finding that warrants further investigation in future studies.

	IndicLID-FTR (8 dim)			IndicLID-BERT (unfreeze 1)			IndicLID (threshold 0.6)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Assamese	37.72	93.55	53.76	66.81	91.21	77.13	72.41	92.77	**81.34**
Bangla	76.63	94.10	84.47	97.12	88.14	92.41	94.94	93.95	**94.44**
Bodo	70.88	98.38	82.40	84.78	99.08	91.37	85.66	99.31	**91.98**
Konkani	24.62	95.72	39.17	38.35	99.32	55.33	40.90	97.75	**57.67**
Gujarati	89.52	78.70	83.76	95.88	85.20	90.23	95.16	86.69	**90.73**
Hindi	65.46	15.68	25.29	76.32	60.40	67.43	77.16	53.32	**63.06**
Kannada	89.66	96.41	92.91	95.79	95.71	95.75	95.29	96.78	**96.03**
Kashmiri	18.74	91.56	31.12	39.45	93.11	55.42	34.80	94.67	**50.90**
Maithili	07.81	38.95	13.01	29.00	41.69	34.21	21.97	43.74	**29.25**
Malayalam	89.75	94.46	92.04	92.19	95.32	93.73	91.33	95.36	**93.30**
Manipuri	64.84	98.87	**78.32**	50.06	98.42	66.36	58.85	99.32	73.91
Marathi	87.21	79.58	83.22	96.35	80.80	87.89	95.86	82.92	**88.92**
Nepali	19.55	82.51	31.61	43.25	93.85	59.21	36.94	93.62	**52.98**
Oriya	41.88	95.70	58.26	64.09	95.51	76.71	62.96	97.27	**76.44**
Punjabi	78.52	37.21	50.49	84.71	64.64	73.32	85.62	62.62	**72.34**
Sanskrit	49.32	96.43	**65.26**	32.55	99.33	49.04	36.88	99.11	53.75
Sindhi	80.00	61.05	69.25	86.39	71.91	78.49	87.88	72.51	**79.46**
Tamil	97.32	90.56	93.82	97.15	93.06	95.06	97.50	92.64	**95.01**
Telugu	94.24	87.68	90.84	95.25	88.76	91.89	95.89	89.50	**92.58**
Urdu	78.88	33.24	46.77	88.53	44.84	59.53	86.87	46.31	**60.41**
Avg	63.13	78.02	63.29	72.70	84.01	74.53	72.74	84.51	**74.72**

Table 10: Precision, recall and F1-score of IndicLID-FTR, IndicLID-BERT and IndicLID roman script model. All scores are calculated on Bhasha-Abhijnaanam roman script test set. **Bold** indicates the best language representation among IndicLID-FTR, IndicLID-BERT and IndicLID roman script model for individual languages.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations (after the conclusion)

☑ A2. Did you discuss any potential risks of your work?
Limitations (after the conclusion)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
We discussed this in Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We discussed this in Section 2 (data) and Section 3 (models).

☑ B1. Did you cite the creators of artifacts you used?
We cited them in Section 2 and 3.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We discuss this in the Ethics Statement (after Limitations)

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We discussed this in Sections 2 and 3.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we used comes from public sources, so no PII is involved. The IndicCorp data we use has already been checked for offensive content. We mention this in the Ethics Statement (after Limitations).

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We discussed this in Section 2.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We discussed this in Section 2.

C ☑ Did you run computational experiments?

We discussed this in Section 3 and Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We discussed this in Section 3.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We discussed this in Section 3, Appendix B, Appendix C and Appendix D.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We discussed this in Section 4.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 2 and 3

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

We discussed this in Section 2 and Appendix A.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
We discussed this in Appendix A.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
We discussed this in Ethics Statement.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We discussed this in Ethics Statement.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
In Ethics Statement