Script-Agnostic Language Identification

Milind Agarwal
George Mason University
magarwa@gmu.edu

Joshua Otten
George Mason University
jotten4@gmu.edu

Antonios Anastasopoulos
George Mason University
jotten4@gmu.edu

Abstract

Language identification is used as the first step in many data collection and crawling efforts because it allows us to sort online text into language-specific buckets. However, many modern languages, such as Konkani, Kashmiri, Punjabi etc., are synchronically written in several scripts. Moreover, languages with different writing systems do not share significant lexical, semantic, and syntactic properties in the neural representation spaces, which is a disadvantage for closely related languages and low-resource languages, especially those from the Indian Subcontinent. To counter this, we propose learning *script-agnostic embeddings* using several different experimental strategies (upscaling, flattening, and script mixing) focusing on four major Dravidian languages (Tamil, Telugu, Kannada, and Malayalam). We find that exposure to a language written in multiple scripts is extremely valuable for script-agnostic language identification, while also maintaining competitive performance on naturally occurring text.

1 Introduction

In many natural language processing tasks, we often need to first identify the source language of a particular text. However, most current methods are unable to account for languages written in non-standard scripts. Many bilingual communities choose to write their minority language in the region's dominant system (such as those in Pakistan, Iran, China), instead of their language's traditional writing system (Ahmadi et al., 2023). It is also common for larger standardized languages to be romanized on the internet, or for languages to have synchronic digraphia (Lehal and Saini, 2014). In this extended abstract, we share preliminary results on script-agnosticism for language identification by analyzing script upscaling and focusing on the four major Dravidian languages: Tamil, Telugu, Kannada, and Malayalam.

2 Method and Experiments

Script Upscaling This method takes a given training example written in one script and "upscales" it into all 4 scripts. Our intuition is that seeing every example in each script will prevent a model from giving weight to any one writing system in its decision-making, forcing it to rely on inherent features of the language. Details available in Appendix A.

Base Model We use `fastText` (Bojanowski et al., 2017) to learn word embeddings, because it provides an efficient way to glean subword information. Without this, we would likely end up with completely separate vectors for each word in a language and would need to implement other strategies to handle out-of-vocabulary (OOV) words.

Dataset We use the FLORES200 dataset (NLLB Team, 2022; Goyal et al., 2021; Guzmán et al., 2019) for training and in-domain testing in all our experiments. For cross-domain tests, we use GlotStoryBooks (Kargaran et al., 2023), UDHR (Kargaran et al., 2023), and MCS-350 (Agarwal et al., 2023).

Evaluation While F1 scores are popular in language identification studies, they are hard to interpret and only have significant advantages when there is a class imbalance in the data distribution. We have selected a training and test set that is evenly distributed and is *not* imbalanced, so we use top-1 accuracy for our evaluation.

Baseline Models Our first baseline model (referred to as `FLORES200`) is trained on the .dev files from FLORES200. We also benchmark with a model pre-trained on Wikipedia, SETimes, and Tatoeba (Joulin et al., 2016). Since this model is state-of-the-art and trained on a large amount of data outside of FLORES200, we use this as a second baseline and will refer to it as `WIKI`.

3 Results

Our model performs quite well on the test sets, with over 96% accuracy (Table 1). Moreover, while it drastically outperformed the FLORES200 baseline on transliterated data, it scored higher on the untransliterated test as well. These results demonstrate that the model was able to correctly disentangle script and language. The WIKI baseline proved superior on the non-transliterated test sets (with a score of 100%), but this matches our expectations, considering that it is a much larger model.

	WIKI		**FLORES200**		**Upscaled**	
Size			3,988		15,952	
	ORI	TRA	ORI	TRA	ORI	TRA
TAM	100	25	94.37	23.59	95.26	95.16
KAN	100	25	92.59	23.15	95.06	95.06
MAL	100	25	86.78	95.85	99.65	99.65
TEL	100	25	94.07	23.52	95.36	95.41
AVG	100	25	95.26	39.26	96.35	96.32

Table 1: This table compares the performance of the Baseline models to the Script-Upscaled model. The SIZE row displays the amount of training data (including transliterations), the ORI column represents the original examples, and the TRA column refers to the examples transliterated to all scripts.

4 Discussion

The results demonstrate that our script-agnostic language identification model performs well above baseline on examples that utilize a non-standard script. We suspect that seeing each example transliterated to every script allows our script-upscaled model to become truly script-agnostic. In the practical setting, our model appears to be a reasonable alternative to current language identification systems, when synchronic digraphia or adversarial writing (writing in a non-traditional script) is expected, especially for South Indian languages.

The WIKI baseline performed the best on the non-transliterated test sets, but this is likely due to its huge amount of training data. It is highly possible that had we trained a Script-Upscaled model on Wikipedia, we would have seen results that matched the WIKI baseline on noiseless data. The large amount of storage and computational power for this endeavor, in addition to potential challenges in transliterating to so many scripts, would have been beyond the scope of our current work. However, now that we have established proof-of-concept, future work will attempt to create fully transliterated WIKI language identification models.

Our approach is relatively straightforward, and requires no more examples than for a standard language identification system. Since transliteration can be done automatically, we essentially propose a data-augmentation process (for complete sentences and within sentences) that results in an ability to classify languages regardless of script. Future work should explore the impact of these script-agnostic embeddings on other downstream tasks, as well as conducting intrinsic evaluation (word analogy and semantic similarity) experiments.

5 Conclusion

In conclusion, we introduce and evaluate a new kind of language identification model that is script-agnostic. It has been shown to outperform the baselines on examples that are not written in the standard script. Our method may provide a reasonable alternative to training language identifiers that can correctly classify text based on the language used, rather than the script in which it is written. We note that our models were trained and evaluated using the four major Dravidian languages - Tamil, Telugu, Malayalam, and Kannada, as a case study. Data loss associated with script conversion and non-phonetic scripts is a likely challenge when we scale our approach to more scripts. Future work would expand to include more languages and scripts, as well as performing tests on the learned embeddings to determine if these would be effective on other downstream tasks.

References

Milind Agarwal, Md Mahfuz Ibn Alam, and Antonios Anastasopoulos. 2023. Limit: Language identification, misidentification, and translation using hierarchical models in 350+ languages.

Sina Ahmadi, Milind Agarwal, and Antonios Anastasopoulos. 2023. PALI: A language identification benchmark for Perso-Arabic scripts. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 78–90, Dubrovnik, Croatia. Association for Computational Linguistics.

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. GlotLID: Language identification for low-resource languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Gurpreet Singh Lehal and Tejinder Singh Saini. 2014. Sangam: A perso-Arabic to indic script machine transliteration model. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 232–239, Goa, India. NLP Association of India.

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 Jeff Wang NLLB Team, Marta R. Costa-jussà. 2022. No language left behind: Scaling human-centered machine translation.

Odunayo Ogundepo, Tajuddeen R Gwadabe, Clara E Rivera, Jonathan H Clark, Sebastian Ruder, David Ifeoluwa Adelani, Bonaventure FP Dossou, Abdou Aziz DIOP, Claytone Sikasote, Gilles Hacheme, et al. 2023. Afriqa: Cross-lingual open-retrieval question answering for african languages. *arXiv preprint arXiv:2305.06897*.

Lisa Yankovskaya, Maali Tars, Andre Tättar, and Mark Fishel. 2023. Machine translation for low-resource finno-ugric languages. In *The 24rd Nordic Conference on Computational Linguistics*.

IPA	ISO	TEL	KAN	MAL	TAM
/ka/	ka	క	ಕ	ക	க
/k^ha/	kha	ఖ	ಖ	ഖ	க₂
/ga/	ga	గ	ಗ	ഗ	க₃
/g^ha/	gha	ఘ	ಘ	ഘ	க₄

Table 2: Tamil has only one letter to represent the above-mentioned 4 sounds common in the other 3 Dravidian languages. So, the transliterator introduces subscripts to differentiate the four sounds in the source script. There are 5 such character series but we only show the *velar* phonemes' series.

A Experimental Details

Upscaling For our script-upscaled model, we first created four training files for each language, where a file would include all of the language's training examples four times–one for each script. Then we concatenated all of these files into one training set. In essence, we allowed our model to assume that a sentence may appear in any of the four writing systems with the same likelihood.

Transliteration We use the Aksharamukha[1] python package to transliterate between our four Dravidian writing systems. Since the library is primarily meant for Indic writing systems, it provides an extremely low-loss transliteration, which is suitable for our purposes. Note that since Tamil has a smaller phonetic inventory than other languages, there may be subscripts introduced during transliteration (see Table 2). We preprocess the Tamil files to remove any such subscripts.

B Out-of-Domain Datasets

A comparison of our models on the clean FLORES200 test set, as well as out-of-domain sets is in Table 3. The FLORES200 BASELINE performs well in-distribution and on similar long-length GLOT and UDHR datasets, but poorly on MCS350 (children's stories domain and shorter sentences). The `WIKI` baseline is better than the FLORES200 baseline across all datasets, showing that is has built a better representation space for the languages.

1. **FLORES200**: n-way parallel dataset consisting of sentences from 842 web articles, translated into a large number of languages (NLLB Team, 2022; Goyal et al., 2021; Guzmán et al., 2019). Each language's

[1] https://pypi.org/project/aksharamukha/

	FLORES200	GLOT	UDHR	MCS350	AVERAGE
Test Set Size	4048	3934	285	15000	5817
BASELINE (FLORES200)	95.26	82.41	79.00	45.34	**75.50**
FASTTEXT (WIKI)	100.00	99.96	100.00	71.75	**92.93**
UPSCALE (16K)	96.35	81.67	77.54	44.79	**75.09**

Table 3: We share two baseline models (trained on FLORES200 and Wikipedia) along with the upscaled model and test them on out of domain data to test domain transfer of the learned embeddings.

example are in the same order, and are separated into `.dev` and `.devtest` files, containing 997 and 1012 sentences, respectively.

2. **GlotStoryBooks**[2]: Open-licensed curated library of books (Kargaran et al., 2023) from a variety of sources in 176 languages (Yankovskaya et al., 2023; Ogundepo et al., 2023). Each sample contains a sentence along with its language identifier and script.

3. **UDHR (Universal Declaration of Human Rights)**: We use Kargaran et al. (2023)'s public domain preprocessed version of the UDHR dataset, where each sample is a paragraph along with a language identifier. The authors removed errors and formatting issues in the original UDHR data and made this clean version available[3].

4. **MCS-350**: Multilingual Children's Stories dataset, released by Agarwal et al. (2023), contains over 50K children's stories curated primarily from two sources - African Storybooks Initiative and Pratham Storyweaver, both open-source story repositories for African and Indian languages respectively. For our experiments, we use the monolingual data files available on the authors' GitHub repository[4].

5. **IndicCorp**[5]: Monolingual, sentence-level corpora for English and 11 Indian languages from the Dravidian and Indo-Aryan families (Kakwani et al., 2020). It consists of 8.8 billion tokens and is sourced mostly from Indian news crawls (articles, blog posts, magazines), though it also takes data from the OSCAR corpus.

C Brief Language Profiles

1. Tamil (`tam`), a Southern-Dravidian language, is spoken by over 80 million people and is an official language in Sri Lanka, the Indian states of Tamil Nadu and Puducherry, and of the Indian Constitution's Eighth Schedule. It is curently most widely written in the Tamil abugida - தமிழ் எழுத்து (*tamizh ezhuttu*).

2. Telugu (`tel`), a South-Central Dravidian language, is spoken by about 100 million people and is the most spoken Dravidian language. It is also an Eighth Schedule language of the Indian Constitution and is official in the Indian states of Andhra Pradesh, Telangana, and Puducherry (Yanam). It is written in Telugu abugida - తెలుగు లిపి (*telugu lipi*)

3. Malayalam, (`mal`), another Southern-Dravidan language is the smallest language from our selection, spoken by about 40 million people in Southern India. It is an Eighth Schedule language and is official in the southernmost Indian state of Kerala. It is written in the Malayalam abugida - മലയാളം അക്ഷരങ്ങൾ (*malayalam aksharangal*).

4. Kannada (`kan`), also a member of the Southern-Dravidian language subfamily, is spoken by about 60 million people, mostly within India. It is an official language of the Indian Constitution's eighth schedule and is the sole official language of Karnataka state. It is widely written in Kannada script, which is closely related to the Telugu script and is also an abugida, but diverged around 1300 CE - ಕನ್ನಡ ಅಕ್ಷರಮಾಲೆ (*kannada aksharamale*).

[2] https://huggingface.co/datasets/cis-lmu/GlotStoryBook

[3] https://huggingface.co/datasets/cis-lmu/udhr-lid

[4] https://github.com/magarw/limit

[5] https://paperswithcode.com/dataset/indiccorp