Transliteration: A Simple Technique For Improving Cross-Lingual Language Modeling

Anonymous authors
Paper under double-blind review

Abstract

While impressive performance in natural language processing tasks has been achieved for many languages by transfer learning from large pretrained multilingual language models, it is limited by the unavailability of large corpora for most languages and the barrier of different scripts. Script difference forces the tokens of two languages to be separated at the input. Thus we hypothesize that transliterating all the languages to the same script can improve the performance of language models. Languages of South Asia and Southeast Asia present a unique opportunity of testing this hypothesis as almost all of the major languages in this region have their own script. Nevertheless, it is possible to transliterate them to a single representation easily. We validate our hypothesis empirically by pretraining ALBERT models on the Indo-Aryan languages available on the OSCAR corpus and measuring the model's performance on the Indo-Aryan subset of the IndicGLUE benchmark. Compared to the non-transliteration-based model, the transliteration-based model (termed XLM-Indic) shows significant improvement on almost all tasks of IndicGLUE. For example, XLM-Indic performed better on News Classification (0.41%), Multiple Choice QA (4.62%), NER (6.66%), and Cloze-Style QA (3.32%). In addition, XLM-Indic establishes new SOTA results for most tasks the on IndicGLUE benchmark while being competitive at the rest. Across the tasks of IndicGLUE, the most underrepresented languages seem to gain the most improvement. For instance, for the NER, XLM-Indic achieves 10%, 35%, and 58.5% better F1-scores for Gujarati, Panjabi, and Oriya languages compared to the current SOTA.

1 Introduction

In the last few years, we have seen impressive advances in many natural language processing tasks. These advances have been primarily led by the availability of large representative corpora and improvement in the architecture of large language models. While improving model architectures, training methods, regularization techniques, etc. can help advance the state of NLP on all languages, the unavailability of large, diverse corpora is the bottleneck for most languages. Even after taking into account the number of speakers, we see a huge disparity among the world languages in the existing multilingual corpora. For example, while the number of Mandarin speakers is comparable to the number of speakers of Indian languages, there are only about 50GB of data for Indian languages in the OSCAR corpus (Ortiz Su'arez et al., 2019), while Mandarin has more than 250GB of data. However, creating representative and inclusive corpora is an ongoing process and is not always possible for many under-resourced languages for many different reasons.

Thus to inclusively advance the state of NLP across languages, it is crucial to train multilingual language models that can extract the most out of existing multilingual corpora. Here we focus on the issue of diverse writing scripts that prevents multilingual models from learning good cross-lingual representations. Previous papers (Pfeiffer et al., 2020) have noticed that under-resourced languages with their own scripts tend to have very few tokens representing them at the tokenizer. As a result, these languages tend to have more UNK tokens, and the words in these languages tend to be more split up by sub-word tokenizers. This issue is particularly exacerbated for the South Asian and Southeast Asian languages

as most of the major languages in this region have their own script. While this issue is evident for Indian languages, we believe there are other examples of closely related languages separated by different scripts.

In this paper, we propose transliteration as a simple technique to improve the capabilities of multilingual language models to extract the most out of existing corpora. We hypothesize that transliterating all the languages to the same script improves the performance of multilingual language models. To test our hypothesis, we trained two ALBERT (Lan et al., 2020) models on the Indo-Aryan languages of the OSCAR corpus. One trained on the original corpus while the other on the transliterated corpus. We then rigorously compared the two model's performance on four different downstream tasks from the IndicGLUE benchmark dataset. Our experiments show that the transliteration model outperformed the baseline model in most tasks.

In summary, our contributions are primarily two-fold:

1. We rigorously compare the baseline model with the transliteration model on benchmark datasets. Our results show that transliteration improves the performance of multilingual language models.
2. We empirically show that transliteration collapses multiple n tokens into a single shared token or n-m tokens where m is defined as $1 \leq m < n$. This improves overall performance and helps underrepresented and under-resourced languages such as Panjabi, Oriya, and Assamese.

2 Motivation and Background

As discussed previously, currently, NLP lacks inclusivity of under-resourced languages. Furthermore, Joshi et al. (2020) showed that the bulk of the allocated resources for NLP research is taken by a few major world languages. Nevertheless, it is essential for the betterment of NLP to include more marginalized and or underrepresented and under-resourced languages (Ruder, 2020). The primary motivation behind the transliteration technique was to find a method to improve the performance of multilingual language models, especially for under-resourced languages. Towards that goal, we chose the Indo-Aryan language family due to two reasons. Firstly, apart from Bengali and Hindi, the rest of the languages are heavily under-resourced and underrepresented. Secondly, many of the South Asian and Southeast Asian languages are intimately connected linguistically and historically and share much vocabulary. However, due to different scripts, it is difficult for language models to exploit this shared information.
Nevertheless, these scripts have shared ancestry from ancient Brahmic script and have similar structures that we can easily transliterate them to a common script. Specifically, the ISO-15919 standard has been designed to do precisely this. Most of these languages have common ancestry through Brahmic scripts (Hockett et al., 1997; Coningham et al., 1996). Many of them also heavily borrow from Sanskrit, and due to its influence, many words are shared between the languages. Due to these languages being phonologically(Littell et al., 2017) and phylogenetically related, a corresponding one-to-one relation exists among many of these languages. Therefore, these languages present a unique opportunity to test the impact of transliteration on the performance of multilingual language models.

In Table 1, we show a text-based example of the mapping between Indo-Aryan languages using transliteration. We chose Hindi and Bengali as representative languages for this example. The first sentence from the universal declaration of human rights was chosen as the example sentence, which is,

Whereas recognition of the inherent dignity and of the equal and inalienable rights of all members of the human family is the foundation of freedom, justice and peace in the world

We can see that both languages share 5 words that match partially and 4 words that match exactly. This is the basis of our hypothesis that instead of having the tokens be separated due to the script barrier, we can collapse them onto a single representative

Table 1: one-to-one correspondence between Bangla and Hindi through transliteration

Language	Sample Sentence
Bangla	মানব পরিবারের সকল সদস্যের সমান ও অবিচ্ছেদ্য অধিকারসমূহ এবং সহজাত মর্যাদার স্বীকৃতিই হচ্ছে বিশ্বে শান্তি, স্বাধীনতা এবং ন্যায়বিচারের ভিত্তি
ISO-15919	mānaba paribārēra sakala sadasyēra samāna ō abicchēdya adhikāra samūha ēbaṁ sahajāta maryādāra sbīkrtii hacchē biśbē śānti, sbādhīnatā ēbaṁ nyāẏabicārēra bhitti
Hindi	मानव परिवार के सभी सदस्यों के जन्मजात गौरव और समान तथा अविच्छिन्न अधिकार की स्वीकृति ही विश्व-शान्ति, न्याय और स्वतन्त्रता की बुनियाद है
ISO-15919	mānava parivāra kē sabhī sadasyōṁ kē janmajāta gaurava aura samāna tathā avicchinna adhikāra kī svīkrt hīviśva-śānti, nyāya aura svatantratā kī buniyāda hai
Color Code	Partial Match Exact Match

token over many languages through transliteration, which should in terms improve the performance of multilingual language models.

Multilingual language models mBERT(Devlin et al., 2019) and XLM-Roberta(Conneau et al., 2020) have shown success in many different tasks; they also have their fair share of limitations(Lauscher et al., 2020; Pires et al., 2019). Performance-wise mBERT also tends to suffer when it comes to under-resourced languages(Wu & Dredze, 2020). On the other hand, transliteration has also been used previously but predominantly for Neural Machine Translation (NMT) tasks(Tennage et al., 2018; Goyal et al., 2020; Durrani et al., 2014; Amrhein & Sennrich, 2020; Song et al., 2020). It has also been used for dependency parsing for code-switching(Bhat et al., 2018) and morphological inflection(Murikinati et al., 2020). However, two recent works come close to our idea, both of which show that transliteration helps improve performance(Khemchandani et al., 2021; Muller et al., 2021). Muller et al. (2021) also showed that transliteration helps under-resourced languages, which alings with our hypothesis.

3 Experiment and Results

3.1 Dataset

We pretrained our model on a subset of the OSCAR corpus. OSCAR corpus is a large multilingual corpus extracted from Common Crawl that contains hundreds of different languages. We use the unshuffled deduplicated version of OSCAR corpus available via Huggingface(Wolf et al., 2020). Due to compute constraints, we only pretrain on the Indo-Aryan languages that use Brahmic scripts. Specifically, we pretrain on Panjabi, Hindi, Bangla, Oriya, Assamese, Gujarati, Marathi Sinhala, Nepali, Sanskrit, Konkani, Maithili, Bihari, and Bishnupriya portion of the OSCAR corpus. We would like to note that the last five languages' corpus size is negligible, but we included them as it was relatively easy to add them to our dataset preparation pipeline.

We evaluate our model on downstream tasks from IndicGLUE (Kakwani et al., 2020) that are publicly available. Thus, we evaluate our model on News category classification, Wikipedia Section-Title Prediction, Cloze-Style QA, and Named Entity Recognition from IndicGLUE. We note that the News Article Headline Prediction dataset from IndicGLUE is not publicly available. In addition, we evaluate our model on other publicly available datasets on which IndicBERT is evaluated. Specifically, we evaluate our model on BBC Hindi News Classification, Soham Bangla News Classification, iNLTKH Headlines Classification, IITP Movie, and Product Review Sentiment Analysis(Akhtar et al., 2016), and MIDAS Discourse Mode Classification(Dhanwal et al., 2020) datasets. We note here that the Amrita Paraphrase datasets(Anand Kumar et al., 2016) are not publicly available.

3.2 Pretraining

Corpus Preparation: Since the OSCAR corpus contains raw text from the Web, we apply a few filtering and normalization. First, we discard entries where the dominant script does not match the language tag provided by the OSCAR corpus. Then we use the Indic-NLP normalizer to normalize the raw text. For XLM-Indic, we then transliterate all the text to ISO-15919 format. We use the Aksharamukha(Aks) library to do the transliteration.

Next, we train two SentencePiece tokenizers (Kudo & Richardson, 2018) on the transliterated and the non-transliterated corpus with a vocabulary size of 50,000. Then we use the trained tokenizer and the sentence-splitter from the IndicNLP library to split long entries from the corpus at sentence boundaries so that no entry may have more than 512 tokens. Finally, we discard short entries (<512 characters) to improve the training efficiency.

ALBERT Model Training: We train two ALBERT models on the non-transliterated and the transliterated corpus. We choose the ALBERT small configuration due to compute constraints. We trained our models on a single TPUv3 VM. Both models were trained using the same hyperparameters. We follow the hyperparameters used in the ALBERT paper except for batch size and learning rate. We used a batch size of 256, which is the highest that fits into TPU memory, whereas the ALBERT paper used a batch size of 4096. As our batch size is $1/16^{\text{th}}$ of the ALBERT paper, we use a learning rate of 1e-3/8, which is approximately $1/16^{\text{th}}$ of the learning rate used in the ALBERT paper (1.76e-2). Additionally, we use the Adam optimizer(Kingma & Ba, 2015) instead of the LAMB optimizer. The rest of the hyperparameters were the same as the ALBERT paper. Specifically, we use a sequence length of 512 with absolute positional encoding, weight decay of 1e-2, warmup steps of 5000, max gradient norm of 1.0, and Adam epsilon of 1e-6. The models were trained for 1M steps. Each model took about 7.5 days to train. From henceforth, unless mentioned otherwise, the non-transliteration model will be called the baseline model.

3.3 Downstream Finetuning

We finetune our model for each downstream task independently. However, where possible, we train a single model for all the languages on a task. The hyperparameter used for each task is reported in the Appendix. We follow the standard practice of finetuning model for sequence classification tasks with classification head on top of the [CLS] token. Similarly, for NER, which is a token classification task, we follow the standard practice of finetuning shared classification head on all tokens. On all tasks, we train models with 9 different seeds and report the average and standard deviation of the metrics. In Table 2 and Table 3, we report performance on IndicGLUE benchmark tasks and on Table 4 on other publicly available datasets. Here we discuss each of the fine-tuning tasks and compare the performance of our model with SOTA models. However, we would like to mention that, in this paper our purpose is not to achieve the SOTA but to understand the impact of transliteration on the performance of language models.

Wikipedia Section Title Prediction This is a multiple-choice question-answer task from IndicGLUE. On this task, we predict the title of a Wikipedia article section from four options. However, we train a sequence classification model instead of a multiple-choice question-answer model for simplicity. Since the task is the same for all languages, we train a single model for all languages. From Table 2, we report the test set accuracy of different models on this task. Here, XLM-Indic achieves state-of-the-art results in all languages.

News Category Classification: This is a news article genre classification task from IndicGLUE. It is available in 5 languages. The dataset sizes vary widely. The Gujarati test set has only 200 samples, whereas the Oriya test set has about 2200 samples. Thus this task helps us test the model's finetuning capability on classification tasks with different dataset sizes. In Table 3, we see that XLM-Indic advances the state-of-the-art on Panjabi, Bangla, and Marathi news classification tasks.

Named Entity Recognition: We use the balanced Wikiann dataset from Rahimi et al. (2019) for this task. We train shared models across languages for this task. In Table 3, we report the F1-scores of different models. We see that XLM-Indic makes significant improvements on Panjabi, Hindi, Bengali, Oriya, and Gujarati named entity recognition.

Article Genre, Sentiment & Discourse Mode Classification: We evaluate our models on some publicly available sequence classification datasets on which IndicBERT also evaluated. Most of these classification tasks are highly skewed. Thus we report F1 scores in addition to accuracy on these tasks. We encourage others to do so in the future. In Table 4, we report the results on this task. We see that XLM-Indic either beats the state-of-the-art or achieves comparable performance on these tasks.

Table 2: Test accuracy on MCQA tasks

Model	pa	hi	bn	or	as	gu	mr	avg
Wikipedia Section Title Prediction								
XLM-R	70.29	76.92	80.91	68.25	56.96	27.39	77.44	65.45
mBERT	72.47	80.12	82.53	22.22	73.42	74.52	80.49	69.40
IndicBERT base	67.39	74.02	80.11	57.14	65.82	68.79	72.56	69.40
IndicBERT large	77.54	77.80	82.66	68.25	56.96	52.23	77.44	70.41
Baseline	74.33±0.83	78.18±0.33	81.18±0.28	74.35±1.2	76.70±0.83	76.37±0.53	79.10±0.84	77.17
XLM-Indic	**77.55±0.61***	**82.24±0.18***	**84.38±0.29***	**81.47±0.99***	**81.74±0.82***	**82.39±0.27***	**82.74±0.52***	**81.78***
Cloze-style QA (Zero Shot)								
IndicBERT base (Our Evaluation)	29.33	30.76	28.45	30.38	29.98	81.50	29.71	37.16
Baseline	31.04	36.72	35.19	34.63	33.92	59.86	36.14	38.21
XLM-Indic	**32.77***	**38.52***	**36.38***	**36.00***	**37.36***	**70.22***	**39.53***	**41.54***

* indicates better compared to SOTA, orange indicates baseline and XLM-Indic are equal and blue indicates XLM-Indic is better

Table 3: Test accuracy on various classification tasks

Model	pa	hi	bn	or	as	gu	mr	avg
News Category Classification								
XLM-R	94.87	-	**98.29**	97.07	-	96.15	96.67	96.61
mBERT	94.87	-	97.71	69.33	-	84.62	96.67	88.64
IndicBERT base	97.44	-	97.14	97.33	-	**100.00**	96.67	97.72
IndicBERT large	94.87	-	97.71	97.60	-	73.08	95.00	91.65
Baseline	96.83±0.19	-	98.14±0.14	98.09±0.16	-	98.80±0.43	99.58±0.25	98.30
XLM-Indic	**97.90±0.17***	-	**97.99±0.22***	**98.77±0.12***	-	**99.40±0.54***	**99.47±0.21***	**98.70***
NER (F1-Score)								
XLM-R	17.86	89.62	92.95	25.00	66.67	55.32	87.86	62.18
mBERT	50.00	86.56	91.81	19.05	**92.31**	68.04	**91.27**	71.29
IndicBERT base	21.43	90.30	93.39	8.69	41.67	54.74	88.71	56.69
IndicBERT large	44.44	86.81	91.85	35.09	43.48	70.21	87.73	65.66
Baseline	76.69±1.5	91.80±0.42	96.39±0.19	84.18±1.8	75.45±1.8	69.10±2.9	88.72±0.40	83.19
XLM-Indic	**85.42±1.9***	**92.93±0.21***	**97.31±0.22***	**93.54±0.58***	**89.06±2.2***	**80.16±0.15***	90.56±0.44	**89.85***

* indicates equal or better compared to SOTA, orange indicates baseline and XLM-Indic are equal and blue indicates XLM-Indic is better

3.4 Zero Shot Capability Testing

We use the Cloze Style Question Answering task to test the zero shot capabilities of our model. On this task, we are given four words to choose from to fill in a gap in a sentence. We use the pretrained masked language model to rank the four choices and predict the top answer. Since a word can be tokenized to multiple tokens by the subword tokenizer, correctly evaluating the model on this task requires special care. Specifically, we have to use the same number of mask tokens as the number of subword tokens that a word gets split into. Then we calculate the probability for the word by multiplying the probability of the subword tokens predicted by the masked language model. We found that on the IndicBERT evaluation code, only a single mask token was used irrespective of the number of subword tokens that a word gets split into. We do not think this is the correct way to evaluate

Table 4: Test accuracies on other public datasets

Language	Dataset	IndicBERT-base	mBERT	XLM-R	SahajBERT	Baseline		XLM-Indic	
						Accuracy	F1-score	Accuracy	F1-Score
Article Genre Classification									
hi	BBC News	74.60	60.55	75.5	-	77.28±1.5	46.47±4.6	**79.14±0.60***	48.59±0.27
bn	Soham Articles	78.45	80.23	87.60	91.97	93.22±0.49	91.12±0.85	**93.89±48***	91.75±0.73
gu	INLTKH Headlines	92.91	89.16	89.83	-	90.41±0.69	88.82±0.73	**90.73±0.75***	89.14±0.88
mr	INLTKH Headlines	94.30	87.50	92.48	-	92.21±0.23	89.23±0.54	**92.04±0.47***	89.09±0.78
Sentiment Analysis									
hi	IITP Product Reviews	71.32	74.57	78.97	-	76.33±0.84	74.04±0.99	77.18±0.77	74.53±0.98
hi	IITP Movie Reviews	59.03	56.77	61.61	-	65.91±2.2	65.26±2.2	**66.34±0.16***	65.87±1.6
Discourse Mode Classification									
hi	MIDAS Discourse	78.44	71.20	79.94	-	78.39±0.33	47.26±6.2	**78.54±0.91***	63.80±3.2

* indicates better compared to SOTA, orange indicates baseline and XLM-Indic are equal and blue indicates XLM-Indic is better

a masked language model on this task. Thus we evaluate IndicBERT with our code and report the results in Table 2.

3.5 Ablation Study With Statistical Testing

As mentioned earlier, we finetuned our model with 9 random seeds. On our baseline model and transliteration model, we report average as well as standard deviation of metric values. On some tasks, the performance of XLM-Indic is so much better than the baseline model that it is easy to make the conclusion. For example, on Wikipedia Section Title Prediction and NER XLM-Indic improves by several percentage points over the baseline model on all languages. Conversely, on other tasks, the difference is minute, and it is not clear whether XLM-Indic is really performing better or not. An example of this is the iNLTKH Gujarati news classification task. Thus, instead of comparing models based on only the average metric value, we conduct statistical tests to determine whether the difference in performance is significant or not.

For News Category Classification, apart from Bengali and Marathi, we observe marginal improvements over the rest of the languages. Our most remarkable performance difference comes from NER, with XLM-Indic performing 6.66% better than the baseline. We see remarkable improvement over Panjabi, Oriya, Assamese, and Gujarati, with Assamese having an accuracy increase of 13.61%.
In general, we see that transliteration helps the underrepresented languages (Panjabi, Oriya, Assamese), while high resource languages (Hindi, Bengali) see marginal improvements. In Table 4, we present the results on other public datasets. As stated earlier, the performance between baseline and XLM-Indic is not discernible for iNLTK Headlines Classification, IITP Product Review Sentiment Analysis, and MIDAS Discourse Mode Classification tasks. The rest of the tasks saw marginal better performance for XLM-Indic.

We performed the Mann–Whitney U test (MW) as our earlier mentioned statistical test, which is, a non-parametric hypothesis test. MW was performed on all downstream tasks except Cloze-style QA. We had three assumptions to verify the results over all the 9 seeds between the baseline and XLM-Indic. The assumptions and their null ($\mathbf{h_0}$) and alternative ($\mathbf{h_a}$) hypotheses are showcased in Table 5. For MW, we set our confidence interval α at 0.05. We rejected the $\mathbf{h_0}$ for p-values $< \alpha$. We start by testing assumption 1 hypotheses. If assumption 1 $\mathbf{h_0}$ could not be rejected, we moved on to assumption 2 hypothesises. If we also could not reject the assumption 2 $\mathbf{h_0}$, we verified the final hypothesises in assumption 3. Failing to reject the $\mathbf{h_0}$ for assumption 3 meant that the models had similar performance. For instance, in Table 6, we can see that for Bengali and Marathi, all 3 assumptions required verification. Our assumptions 1, 2 and 3 can also be seen as right-tailed, left-tailed, and two-tailed hypothesis tests, respectively. The rest of the p-values for other tasks are reported in the Appendix.

Table 5: Mann Whiteney U test assumptions and hypotheses

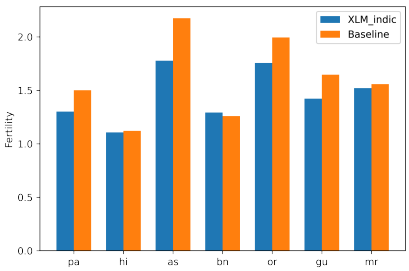
Assumption 1 (Primary hypothesis)	
$\mathbf{h_0}$**:**	XLM-Indic $\leq$ baseline.
$\mathbf{h_a}$**:**	XLM-Indic $>$ baseline.
Assumption 2 (If we can not reject $\mathbf{h_0}$ for assumption 1)	
$\mathbf{h_0}$**:**	XLM-Indic $\geq$ baseline.
$\mathbf{h_a}$**:**	XLM-Indic $<$ baseline.
Assumption 3 (If we can not reject $\mathbf{h_0}$ for assumption 2)	
$\mathbf{h_0}$**:**	XLM-Indic $=$ baseline.
$\mathbf{h_a}$**:**	XLM-Indic $\neq$ to baseline.

Table 6: Mann Whitney U test |p-value < 0.05|

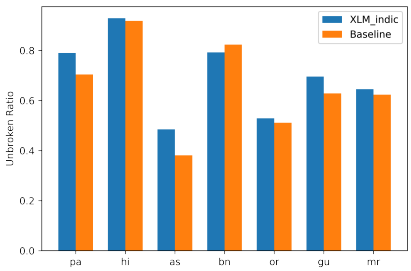
Model	**pa**	**hi**	**bn**	**or**	**as**	**gu**	**mr**
News category classification							
$\mathbf{h_0}$**:** Assumption 1	0.0001	-	0.9232	-	0.0154	0.0073	0.9296
$\mathbf{h_0}$**:** Assumption 2	-	-	0.0905	-	-	-	0.08416
$\mathbf{h_0}$**:** Assumption 3	-	-	0.181	-	-	-	0.1683

4 Why Transliteration Works

Closely related languages often share a lot of vocabulary. Multilingual languages can pick up on these shared words and learn cross-lingual word representations. Pivoting on these shared words, the model can learn the connection between other words. Often a word is rarely used in one language but is frequently used in another language. Multilingual models can learn good representation for the word from the corpus of the later language and improve performance on the former language. Even if the shared words only partially match their spelling, subword tokenizers can often exploit the subword similarity.



(a) Subword Fertility.



(b) Unbroken Ratio.

Figure 1: Subword fertility (lower is better) and unbroken ratio (higher is better)

However, all of the mechanisms through which multilingual models can learn good cross-lingual representations stop working if the closely related languages do not share the same writing scripts. Since otherwise same words written in different scripts get mapped to different tokens, the model faces tremendous difficulty in learning all the associations that are present in the data. When we transliterate all the languages to the same script, these

problems disappear. Indeed our experimental results described in section 3 conclusively show the benefit of transliteration.

However, we can also show that transliteration collapses these shared words to the same token by measuring the subword fertility (average number of tokens per word) and the ratio of words unbroken by the tokenizer. From Figure 1, we can see that transliteration reduces the splitting of words. This indicates that many words that were represented by different tokens in the non-transliterated model are represented by a single token in the transliteration model.

5 Conclusion and Future Work

We have conducted a rigorous empirical analysis to quantify the impact of transliteration on multilingual language modeling. We have shown that transliteration as a preprocessing technique improves the performance of multilingual language models. We also showed that transliteration collapses multiple n tokens into a single shared token or n-m tokens where m is defined as $1 \leq m < n$, which removes the script barrier between morphologically similar languages and helps under-resourced languages achieve better results.
Due to compute constraints, we had to use small models. Also, we had to restrict ourselves to a small pretraining corpus and only the Indo-Aryan languages. In the future, we would like to train larger models on a larger corpus such as mC4 and a more diverse set of languages. We would also like to train character-based language models that can exploit the full potential of inter-language vocabulary sharing exposed by transliteration.

6 Reproducibility Statement

We have taken great care to ensure our results are reproducible. The dataset details are provided in subsection 4.1, and the data processing steps and pretraining details along with hyperparameters are described in subsubsection 4.2. Finetuning hyperparameters are provided in appendix A.1. We have also provided our code as supplementary material.

References

Aksharamukha transliteration tool. `https://aksharamukha.appspot.com/converter`. Accessed: 2021-10-04.

Md. Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. Aspect based sentiment analysis: Category detection and sentiment classification for hindi. In *proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016), April 3-9, 2016, Konya, Turkey*, 2016.

Chantal Amrhein and Rico Sennrich. On Romanization for model transfer between scripts in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2461–2469, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.223. URL `https://aclanthology.org/2020.findings-emnlp.223`.

M. Anand Kumar, S. Singh, B. Kavirajan, and K.P. Soman. DPIL@FIRE 2016: Overview of shared task on detecting paraphrases in Indian Languages (DPIL). In *CEUR Workshop Proceedings*, 2016.

Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. Universal Dependency parsing for Hindi-English code-switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 987–998, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1090. URL `https://aclanthology.org/N18-1090`.

R.A.E. Coningham, F.R. Allchin, C.M. Batt, and D. Lucy. Passage to india? anuradhapura and the early use of the brahmi script. *Cambridge Archaeological Journal*, 6(1): 73–97, April 1996. doi: 10.1017/s0959774300001608. URL https://doi.org/10.1017/s0959774300001608.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Swapnil Dhanwal, Hritwik Dutta, Hitesh Nankani, Nilay Shrivastava, Y. Kumar, Junyi Jessy Li, Debanjan Mahata, Rakesh Gosangi, Haimin Zhang, R. R. Shah, and Amanda Stent. An annotated dataset of discourse modes in hindi stories. In *LREC*, 2020.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. Integrating an unsupervised transliteration model into statistical machine translation. In *EACL*, 2014.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 162–168, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-srw.22. URL https://aclanthology.org/2020.acl-srw.22.

Charles F. Hockett, Peter T. Daniels, and William Bright. The world's writing systems. *Language*, 73(2):379, June 1997. doi: 10.2307/416027. URL https://doi.org/10.2307/416027.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 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*, pp. 6282–6293, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4948–4961, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.445. URL https://aclanthology.org/2020.findings-emnlp.445.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. Exploiting language relatedness for low web-resource language model adaptation: An Indic languages study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1312–1323, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.105. URL https://aclanthology.org/2021.acl-long.105.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, 2018.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942, 2020.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4483–4499, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.363. URL https://aclanthology.org/2020.emnlp-main.363.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2002.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 448–462, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.38. URL https://aclanthology.org/2021.naacl-main.38.

Nikitha Murikinati, Antonios Anastasopoulos, and Graham Neubig. Transliteration for cross-lingual morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 189–197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.sigmorphon-1.22. URL https://aclanthology.org/2020.sigmorphon-1.22.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pp. 9 – 16, Mannheim, 2019. Leibniz-Institut f"ur Deutsche Sprache. doi: 10.14618/ids-pub-9021. URL http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Unks everywhere: Adapting multilingual language models to new scripts. *arXiv preprint arXiv:2012.15562*, 2020.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL https://aclanthology.org/P19-1493.

Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 151–164, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1015.

Sebastian Ruder. Why You Should Do NLP Beyond English. http://ruder.io/nlp-beyond-english, 2020.

Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. Pre-training via leveraging assisting languages for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 279–285, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-srw.37. URL https://aclanthology.org/2020.acl-srw.37.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328. JMLR.org, 2017.

Pasindu Tennage, Achini Herath, Malith Thilakarathne, Prabath Sandaruwan, and Surangika Ranathunga. Transliteration and byte pair encoding to improve tamil to sinhala neural machine translation. *2018 Moratuwa Engineering Research Conference (MERCon)*, pp. 390–395, 2018.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Shijie Wu and Mark Dredze. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 120–130, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.16. URL https://aclanthology.org/2020.repl4nlp-1.16.

A Appendix

A.1 Downstream Hyperparameters

All of our hyperparameters for downstream tasks are presented in Table

Table 7: Hyperparameters for all tasks

Task	TPU	Batch Size	Learning Rate	Weight Decay	Dropout	Epochs	Warmup Ratio
News Category Classification	False	16	2e-5	0.01	0.1	20	0.10
Wikipedia Section-Title Prediction	True	256	2e-5	0.01	0.1	3	0.10
Named Entity Recognition	True	512	2e-5	0.01	0.1	20	0.10
BBC Hindi News Classification	False	16	2e-5	0.01	0.1	20	0.10
Soham Bangla NewsClassification	False	16	2e-5	0.01	0.1	8	0.10
iNLTKH Headlines Classification	False	256	2e-5	0.01	0.1	20	0.10
IITP Movie Review	False	64	5e-5	0.01	0.25	20	0.10
IITP Product Review	False	16	5e-5	0.01	0.5	20	0.10
MIDAS Discourse Mod	False	32	2e-5	0.01	0.5	20	0.10

A.2 Case Study on Soham Bangla News Classification:

In order to further probe the difference between classification decision of XLM-Indic and baseline model, we performed multiple out of distribution adversarial examples along with Layer Integrated Gradients method (Sundararajan et al., 2017) to check the input features attribution. The experiments were performed using baseline and XLM-indic Soham Bangla News Classification. Soham Bangla News Classification was chosen due to it being a multi-class classification task. It has six classes which are 'entertainment', 'international', 'kolkata', 'national', 'sport' and 'state'. The adversarial examples were created by mixing class dependant words. For example, "They won the award for their acting performance." should be classified as entertainment. However, "They won the penalty by acting injured." should be classified as sports instead of entertainment. In our experiments, XLM-Indic tended to perform better compared to baseline and capture more correct class dependant words instead of the adversarial ones. However, providing certain class dependant keyword (i.e. 'police', 'political, 'actor', 'footballer') let the baseline model fix its misclassification.

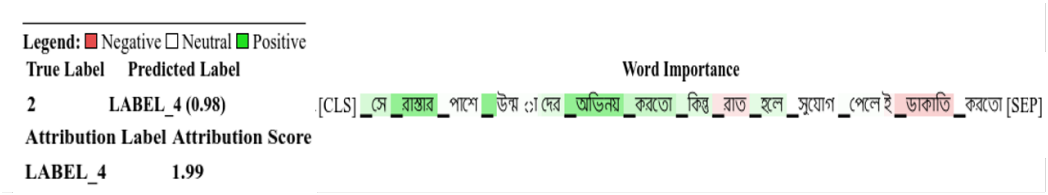


Figure 2: Feature importance for baseline

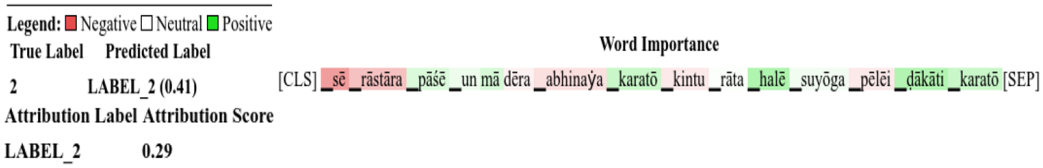


Figure 3: Feature importance for XLM-Indic

We provide one such example below.
We used the example, □□সে রাস্তার পাশে উন্মাদের অভিনয় করতো কিন্তু রাত হলে সুযোগ পেলেই ডাকাতি করতো।", which translates to, "He used to act insane on the side of the road however at night he would rob whenever he got the chance". The true label for this example under Soham News Classification should be 'state'. Which is label 2 in the dataset. Here act is used as a class dependant adversarial word to deviate the model towards 'entertainment'. We would like the model to attribute the correct word 'rob' instead. Figure 4 shows the word attributions for the baseline model. We can see that the baseline classified the sentence as 'entertainment' (label 4) and put positive attribution on অভিনয় , which means act. On the other hand it put negative attribution on ডাকাতি, which means rob. However, XLM-Indic correctly classified the sentence and put the expected positive and negative attributions on 'rob' and 'act' respectively. Nevertheless, as stated earlier, allowing certain class dependant keywords fixes the baselines misclassification. For example, if we were to add 'Police report states that' in front of our provided example, the baseline model would have given the correct class.

A.3 Mann Whitney U test p-values

The Mann Whiteney U test p-values for ablation study on Wikipedia Section Title Prediction (WSTP) and NER is given in Table 8

Table 8: Mann Whitney U test |p-value < 0.05| NER and WSTP

Model	**pa**	**hi**	**bn**	**or**	**as**	**gu**	**mr**
Wikipedia Section Title Prediction							
$\mathbf{h_0}$**:** Assumption 1	0.0002	0.0002	0.0001	0.0002	0.0002	0.0002	0.0002
NER							
$\mathbf{h_0}$**:** Assumption 1	0.0002	0.0003	0.0001	0.0002	0.0002	0.0002	0.0002

The Mann Whiteney U test p-values for ablation study on the other public datasets is given in Table 9

Table 9: Mann Whitney U test |p-value < 0.05| other public datasets

Language	Dataset	$\mathbf{h_0}$ Assumption 1		$\mathbf{h_0}$ Assumption 2		$\mathbf{h_0}$ Assumption 3	
		Accuracy	F1-score	Accuracy	F1-Score	Accuracy	F1-Score
Article Genre Classification							
hi	BBC News	0.0002	0.2181	-	0.8067	-	0.4363
bn	Soham Articles	0.0045	0.0252	-	-	-	-
gu	INLTKH Headlines	0.3124	0.2729	0.7183	0.7553	0.5457	0.6249
mr	INLTKH Headlines	0.8468	0.6976	0.1752	0.3332	0.3503	0.6665
Sentiment Analysis							
hi	IITP Product Reviews	0.0205	0.1933	-	0.8299	-	3865
hi	IITP Movie Reviews	0.447	0.2729	0.5878	0.7553	0.8941	0.5457
Discourse Mode Classification							
hi	MIDAS Discourse	0.6552	0.00002	0.378	-	0.7561	-

A.4 Pretraining loss curves

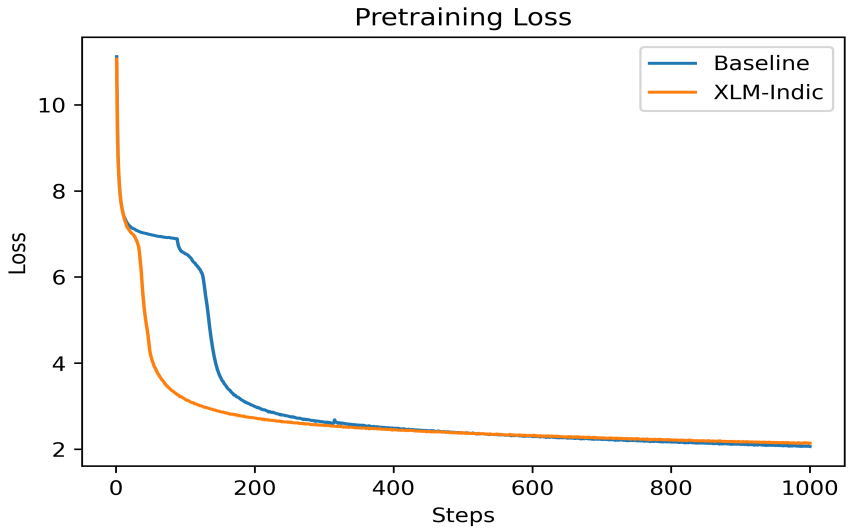


Figure 4: Pretraining loss: Baseline vs Transliteration model