IndicBART: A Pre-trained Model for Indic Natural Language Generation

Anonymous ACL submission

Abstract

We study pre-trained sequence-to-sequence model for a specific-language family with a focus on Indic languages. We present IndicBART, a multilingual, sequence-to-sequence pre-trained model focusing on 11 Indic languages and English. IndicBART utilizes the orthographic similarity between Indic scripts to improve transfer learning between similar Indic languages. We evaluate IndicBART on two NLG tasks: Neural Machine Translation (NMT) and extreme summarization. Our experiments on NMT and extreme summarization show that a language family-specific model like IndicBART is competitive with large pre-trained models like mBART50 despite being significantly smaller. It also performs well on very low-resource translation scenarios: languages not included in pre-training or fine-tuning. Script sharing, multilingual training and better utilization of limited model capacity contribute to the good performance of the compact IndicBART model.

1 Introduction

Recently, there has been significant progress in deep learning based natural language generation (NLG) for machine translation, abstractive summarization, data-to-text generation etc. due to the adoption of attention-based sequence-to-sequence (S2S) models (conditional language models) (Wu et al., 2016; Paulus et al., 2018; Puduppully et al., 2019). Pre-trained S2S models have been shown to be useful to improve performance on various NLG tasks (Rothe et al., 2020; Kale and Rastogi, 2020; Lewis et al., 2020). Specifically, multilingual pre-trained S2S models jointly trained on monolingual corpora from multiple languages such as mBART25 (Liu et al., 2020), mBART50 (Tang et al., 2020a) and mT5 (Xue et al., 2021) have seen increased adoption and low-resource languages have benefitted from cross-lingual transfer. However, these massively multilingual massive (M3) models have major limitations. They serve only a few of the world's languages (<100 languages), the pre-training corpora are dominated by high-resource languages, the vocabulary representation for low-resource languages is inadequate, and the models are large, making them expensive and slow to train, fine-tune and decode.

An alternative approach is to build pre-trained S2S models for a group of related languages. Previous work has shown the benefits of pre-trained language models as well as NMT models that cater to a set of related languages (Kakwani et al., 2020; Tan et al., 2019; Khanuja et al., 2021). However, such a study on multilingual pre-trained S2S models is missing in the literature. In this work, we address this gap in the literature by studying multilingual pre-trained S2S models for Indic languages.

The result of this study is *IndicBART*, a multilingual pre-trained sequence to sequence model specifically trained for Indic languages, which are spoken by more than a billion users[1]. It **supports English and 11 Indian languages** including 7 Indo-Aryan (Assamese, Bengali, Gujarati, Hindi, Marathi, Odiya, Punjabi) and 4 Dravidian (Kannada, Malayalam, Tamil, Telugu) languages. Of these, mBART25, mBART50 and mT5 support only 2, 7 and 9 languages respectively. It is a **compact model with just 244M parameters**, which is much smaller than the M3 models such as mBART50 and mT5(-base) which contain 611M and 580M parameters respectively. We also propose a variant of IndicBART, i.e. **IndicALBART**, that is highly compact with just **97M parameters**.

We compare IndicBART with M3 models on two downstream generation tasks: machine translation and extreme summarization (Narayan et al., 2018). The results indicate that IndicBART is competitive or better by up to 2 BLEU/ROUGE compared to M3 models like mBART50. IndicBART also

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

performs well in the following zeroshot scenarios: (a) on languages not included in pre-training, and (b) languages for which there is no finetuning data.

The following aspects of the IndicBART model contribute to its strong performance and increased language coverage within the Indic family vis-à-vis M3 models, while being highly compact:

1. It is trained on a smaller set of related languages which reduces model capacity requirements. Moreover, available model capacity is effectively utilized since transfer learning works when languages share linguistic features and data represents shared topical themes.

2. It is trained on the largest publicly available Indic language corpora, IndicCorp (Kakwani et al., 2020), which includes large, high-quality news crawls for Indian languages as well as English content from Indian websites - thus being representative of Indian English and topics.

3. We utilize the orthographic similarity between Indic scripts (Kunchukuttan et al., 2018) to map all the Indic language data to a single script, effectively reducing the number of scripts from 9 to 1 (each script having approximately 50 characters). This increases the shared subwords in the vocabulary, and we observe that single script models enable better cross-lingual transfer while finetuning. Since subwords embeddings consume a significant fraction of the parameter space, single script models also better utilize available vocabulary budget.

4. Extremely compressed pre-trained S2S models (IndicALBART) suitable for deployment can be trained by sharing parameters across layers of the transformer layers. For related languages, we show compressed pre-trained models are competitive with full models on downstream tasks when finetuned on distilled data.

The IndicBART model and its variants will be made available under an MIT license to spur further innovation in NLG for Indic languages and study of pre-trained S2S models for related languages.

2 Related Work

Pre-trained models. Pre-trained models learnt using self-supervised objectives and large monolingual corpora have contributed to rapid advances in NLU (Devlin et al., 2019) and NLG (Lewis et al., 2020). Following initial work on English pre-trained models, multilingual pre-trained models have been proposed for NLU (Devlin et al., 2019; Conneau et al., 2020) as well as NLG (Liu et al., 2020; Tang et al., 2020a; Xue et al., 2021) supporting around 100 languages. These pre-trained M3 models have proven to be very useful in improving NLG performance in low-resource settings, especially for applications other than translation.

Language family-specific models. The proposed IndicBART model is also a multilingual pre-trained S2S model similar in architecture and training to mBART. However, in contrast to mBART and mT5, the proposed IndicBART caters specifically to Indic languages. While language-family specific NLU language models like IndicBERT (Kakwani et al., 2020) and MuRIL (Khanuja et al., 2021) and NMT models (Tan et al., 2019) have been proposed, ours is one of the first efforts to create a pre-trained S2S model for a specific language family (and the first for Indic languages). AfroMT (Reid et al., 2021) is a concurrent effort focussed on African languages and low monolingual corpora scenarios. While AfroMT effort is focussed on MT, we investigate IndicBART on another NLG task as well - abstractive summarization. Interestingly, the publicly available family-specific language models (IndicBERT and MuRIL) both cater to Indic languages, pointing to perceived need for Indic language specific models.

Language relatedness. Language-family specific models are motivated from previous work that emphasizes the role of language relatedness in cross-lingual transfer for NMT (Nguyen and Chiang, 2017; Dabre et al., 2017; Aharoni et al., 2019; Kudugunta et al., 2019; Dabre et al., 2020) and NLU (Kakwani et al., 2020; Khemchandani et al., 2021; Dhamecha et al., 2021). We use a single script for representing Indic data since orthographic similarity between Indic languages has been utilized to represent data in a common script and improve cross-lingual transfer for machine transliteration (Kunchukuttan et al., 2018), machine translation (Dabre et al., 2018; Goyal et al., 2020; Ramesh et al., 2021) and NLU (Khemchandani et al., 2021; Dhamecha et al., 2021).

Parameter Sharing and Distillation. Parameter sharing across layers has shown promise for NMT (Dabre and Fujita, 2019) and pre-trained LMs (Lan et al., 2020) in building compressed models while maintaining end-task performance. The IndicALBART model proposed in this work is the first model to explore parameter-sharing across layers for pre-trained S2S models. For NMT models trained from scratch, sequence-to-sequence distillation (Kim and Rush, 2016) has been shown as

an effective way to transfer knowledge to smaller models, while training large models on distilled data (a form of self-training) has been shown to improve translation quality (Dabre and Fujita, 2020). Our results indicate that these results hold when fine-tuning on pre-trained S2S models as well.

3 IndicBART

The **IndicBART** model is conceptually based on the mBART25/50 model family which are Transformer models (Vaswani et al., 2017) trained on monolingual corpora with masked span reconstruction objective. We refer the readers to the mBART literature (Lewis et al., 2020; Liu et al., 2020) for architectural details and highlight specific details and differences from the mBART25/50 setup.

3.1 Design Considerations for IndicBART

Considerations that drove our model choices are:
Compactness: The model should be compact given our focus on a smaller set of related languages, as well as to accelerate training and fine-tuning. Such a model will be usable by a larger base of users with limited computational resources.
Content Relevance: In addition to Indian languages, we include English since transfer-learning from English is a natural use case, and English is widely used in the Indian subcontinent. We also use English content from the Indian subcontinent to reflect relevant content.
Leveraging Relatedness: We utilize orthographic similarity between Indian languages, most of which use abugida scripts derived from the Brahmi script. The logical character set has high overlaps, though each script has its own code-point range in the Unicode standard. We map all the data to Devanagari, enabling better transfer learning with a more compact vocabulary compared to mBART.

3.2 Model and Training Details

IndicBART uses (N=) 6 encoder and decoder layers with hidden and filter sizes of 1024 and 4096, respectively, and 16 attention heads (244M parameters). Similar to mBART, we mask (p=)35% of the words in each sentence by randomly sampling a span length according to a Poisson distribution ($\lambda = 3.5$). We use dropouts of 0.1, label smoothing of 0.1, Adam optimizer with a maximum learning rate of 0.001, weight decay of 0.00001, linear learning rate warmup and decay with 16,000 warmup steps, batch sizes of 4096 tokens. We train for 750,000 iterations on 48 NVIDIA V-100 GPUs, corresponding to roughly 2 epochs, taking around 5 days[2]. In comparison, mBART25/50 models need much longer time (2+ weeks) on 256 GPUs.

To explore more compressed pre-trained models, we train **IndicALBART**, a variant of IndicBART with cross-layer parameter sharing, i.e., sharing parameters across layers. For ablation studies on the impact of single script representation we also train a variant of IndicBART with a 64K vocabulary using the original scripts, which we call separate script IndicBART (SSIndicBART).

The models have been trained with the YANMTT toolkit[3] (Dabre and Sumita, 2021) which is based on the mBART implementation of the HuggingFace Transformers library (Wolf et al., 2020).

3.3 Training Data and Pre-processing

We train the IndicBART model on the IndicCorp (IC) dataset (Kakwani et al., 2020) which contains 11 Indic languages and English. The Indic languages are: Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa), Tamil (ta) and Telugu (te). The corpora statistics are mentioned in Table 7 of the appendix. We train the model on a total of approx. 450 million sentences and 9 billion tokens where corpora sizes are balanced with temperature (T=5) based sampling (Arivazhagan et al., 2019). All the Indic language data is represented in a single script, i.e., the Devanagari script using the IndicNLP library[4] (Kunchukuttan, 2020). We use a vocabulary of 64K subwords learned using SentencePiece (Kudo, 2018; Kudo and Richardson, 2018) on randomly sampled 1M raw sentences from the IndicCorp for each language, for a total of 12M sentences. The model is trained at the sentence-level, unlike the mBART50 model, which is trained on contiguous text chunks potentially spanning multiple sentences.

4 Experiments: NMT

Machine Translation is a standard, popular, cross-lingual generation task for which various pre-trained models are evaluated. We compare IndicBART and its variants with mBART50 which should be the most directly comparable model. We

[2]Longer training was limited by the availability of many GPUs simultaneously.
[3]https://github.com/prajdabre/yanmtt
[4]https://github.com/anoopkunchukuttan/indic_nlp_library

study their performance in: (a) low-resource, (b) multilingual and (c) zero-shot training settings.

4.1 Models Compared

We study IndicBART via the following models:

Models trained from scratch: We train bilingual (Bi) as well as multilingual many-to-one (M2O) and one-to-many (O2M) transformer models.

Fine-tuned models: We fine-tune mBART50 (MB50), IndicBART (IB) and its variants namely IndicALBART (IALB) and separate script IndicBART (SSIB). The type of fine-tuning is indicated by +type which can be Bi, O2M or M2O. If needed, the corpus is indicated by +corpus.

Distilled models: We use the multilingually fine-tuned IndicBART model and translate the training data source sentences which yields distillation data (Kim and Rush, 2016). We use this data to train M2O and O2M models from scratch, as well as by fine-tuning on mBART50, IndicBART and IndicALBART. This was motivated by Dabre and Fujita (2020) who show that the distillation data generated using models employing transfer learning significantly improves the performance of compact models for low-resource languages.

4.2 Datasets and Preprocessing

The statistics of training corpora are in Table 7 in the appendix.

Training: For a low-resource setting (LR), we use the PMI subset (Haddow and Kirefu, 2020) of the WAT 2021 MultiIndicMT[5] (Nakazawa et al., 2021) training set for finetuning. This represents an extremely low-resource parallel corpus setting where we expect IndicBART to be the most helpful. We experiment with extending the PMI data (326K pairs) with the CVIT-PIB (henceforth PIB: 930K pairs) data (Siripragrada et al., 2020) which is similar in domain to the former. We also use the high-resource, general domain Samanantar corpus (Ramesh et al., 2021) (46.2M pairs) to compare with the generalization capabilities of pre-trained models which are fine-tuned with small corpora (PMI, PIB).

Testing: We use the WAT 2021 MultiIndicMT testset and the FLORES101 devtest (Goyal et al., 2021) for evaluation of our models. Both these testsets are n-way parallel (2,390 and 1,012 sentences respectively). The WAT 2021 testset shares the same domain as the training set. The FLORES devtest comes from a different, general domain. We rely on the FLORES dataset to evaluate performance of models trained on the PMI/PIB domain on a more general domain.

Validation: We use the WAT2021 development set of 1,000 sentences.

Preprocessing: For IndicBART and IndicALBART, we use the Indic NLP library to convert the Indic side of the parallel data to the Devanagari script. For mBART50, only Kannada, Punjabi and Oriya scripts are converted to Devanagari as mBART50 does not support these languages. Results for these are italicized. For separate script IndicBART we do not do script conversion.

With this setup, we study the benefits of pre-training in low-resource settings (finetuned on PMI and PIB) and compare it with high-resource settings (trained on Samanantar) on in-domain (WAT2021) and general (FLORES) testsets. Unless explicitly mentioned, our models are assumed to be trained/fine-tuned/distilled with the PMI training data.

4.3 Model Training Settings

We use a single GPU for bilingual and 8 GPUs for multilingual models, all of which are Transformers. Multilingual models are trained using the approach in Johnson et al. (2017) where corpora for various language pairs are first balanced according to their size, then concatenated after appending target language indicator tokens, and finally fed to the NMT model for training. Wherever possible and applicable, we tuned hyperparameters such as hidden sizes, dropout, label smoothing, warmup, tokens per batch, per GPU, learning rate and weight decay. The ADAM optimizer was used. We train our models till convergence on the development set BLEU scores (Papineni et al., 2002). We decode train/tests sets using beam search with a beam of size 4 and a length penalty of 0.8. We report the BLEU scores on the decoded results computed using sacreBLEU[6] (Post, 2018). For additional details, refer to section B in the appendix.

4.4 Comparison of Pre-trained Models

We first describe the main results of using IndicBART and its variants for machine translation and compare it with other relevant models. Table 1 shows results for models trained on the PMI corpus and evaluated on the WAT21 testset.

[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual

[6] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.5.1

Model	#Params	bn	gu	hi	kn	ml	mr	or	pa	ta	te
XX-En											
Bilingual Models											
Bi	78M	13.5	27.4	30.9	22.5	16.5	18.4	18.4	27.1	17.1	16.5
MB50+Bi	611M	23.2	35.4	**38.3**	*26.8*	**29.2**	**27.7**	*27.8*	*35.8*	**27.1**	**30.8**
IB+Bi	244M	**23.6**	**35.5**	36.8	**31.6**	27.9	26.8	**28.3**	**36.3**	27.0	29.9
Multilingual Models											
M2O	78M	18.9	24.8	27.8	23.8	21.6	20.7	21.2	26.4	20.6	21.8
MB50+M2O	611M	**24.8**	**33.9**	36.8	*30.1*	**28.8**	28.1	*27.5*	*34.5*	27.0	29.2
IB+M2O	244M	**24.8**	**33.9**	**37.2**	**32.4**	28.5	**28.5**	**28.8**	**35.7**	**27.3**	**29.5**
IALB+M2O	97M	23.1	33.2	34.4	29.5	27.1	27.0	27.3	34.1	25.2	27.4
Distilled Large Models											
MB50+M2O	611M	26.1	35.9	38.3	*32.9*	29.6	29.3	*30.1*	*37.1*	28.5	31.7
IB+M2O	244M	26.0	35.9	38.0	33.7	29.9	29.4	30.3	37.4	28.4	31.6
Distilled Compact Models											
M2O	78M	23.6	33.3	36.0	30.2	26.0	26.9	27.7	34.0	25.6	27.8
IAIB+M2O	97M	**24.9**	**34.4**	**36.6**	**31.9**	**27.7**	**28.1**	**28.6**	**35.5**	**26.5**	**29.0**
En-XX											
Bilingual Models											
Bi	78M	4.5	17.9	21.7	12.1	3.9	10.0	9.2	17.9	7.2	2.1
MB50+Bi	611M	**8.6**	23.5	**27.0**	*17.4*	**6.0**	**15.8**	*11.6*	*24.5*	**11.2**	3.3
IB+Bi	244M	8.2	**23.6**	26.9	**17.7**	**6.0**	**15.8**	**11.8**	**25.1**	10.8	**3.6**
Multilingual Models											
O2M	78M	7.4	22.5	25.9	16.2	5.6	14.7	11.4	21.9	10.0	2.7
MB50+O2M	611M	8.9	22.8	**27.5**	*18.1*	6.5	16.3	*12.0*	*25.1*	**11.6**	**3.7**
IB+O2M	244M	**9.1**	**24.0**	27.3	**18.5**	**6.7**	**16.7**	**12.9**	**26.4**	**11.6**	**3.7**
IALB+O2M	97M	8.1	22.3	26.3	17.0	5.8	15.3	11.6	24.2	10.5	3.2
Distilled Large Models											
MB50+O2M	611M	**9.4**	24.5	27.5	*17.5*	6.1	16.4	*12.8*	*26.3*	11.6	2.9
IB+O2M	244M	9.3	**25.0**	**28.2**	**19.2**	**6.7**	**17.0**	**13.2**	**26.5**	**11.8**	**3.7**
Distilled Compact Models											
O2M	78M	**8.9**	**24.1**	**27.5**	**18.2**	**6.3**	16.0	12.5	**25.6**	11.0	**3.2**
IAIB+O2M	97M	**8.9**	23.4	27.2	17.8	**6.3**	**16.2**	**12.7**	25.3	**11.3**	3.1

Table 1: Comparison of IndicBART with other models. Scores are reported on the WAT 2021 test set.

Language specific models are compact and competitive: Considering bilingual models, IndicBART outperforms models trained from scratch and gives competitive results when compared to mBART50. For Indic to English translation, mBART50 tends to be better but this is not surprising because it is trained on far larger amounts of English data in addition to being almost 3 times larger than IndicBART. For English to Indic translation, both models tend to give similar scores. In the case of multilingual models, IndicBART is, once again, vastly better than its counterpart trained from scratch and when compared to mBART50 the gap which existed in case of bilingual settings disappears and sometimes reverses in favor of IndicBART. In both cases, IndicBART outperforms mBART50 for Kannada, Punjabi and Oriya which the latter is not trained for. This shows that having a compact language family specific model can be competitive with if not better than a general purpose model trained on a larger number of languages while only having one-third the number of parameters as the latter.

Extreme compression has its downside: Comparing the performance of IndicBART and mBART50 against IndicALBART in multilingual settings, it seems that a 60% and 84% reduction of parameters, respectively, has a negative impact on the translation quality, which results in drops of up to 3 BLEU. However, this may be considered as a

Model	bn	hi	ml	or	ta
	XX-En				
IB+M2O	**24.8**	**37.2**	**28.5**	**28.8**	**27.3**
SSIB+M2O	24.1	35.5	27.9	28.1	26.9
	En-XX				
IB+O2M	**9.1**	**27.3**	**6.7**	**16.9**	**11.6**
SSIB+O2M	9.3	**27.3**	6.2	16.6	11.4

Table 2: Ablation studies on the impact of multilingualism and script unification on downstream performance of IndicBART. Scores are on the WAT 2021 test set.

Model	bn	hi	ml	or	ta
	Test Set: WAT 2021				
IB+PMI	24.8	37.2	28.5	28.8	27.3
IB+PMI+PIB	**28.9**	**41.7**	**33.2**	**33.2**	**32.0**
Samanantar	27.9	41.8	32.7	32.9	31.2
IB+Samanantar	27.1	41.0	31.6	32.3	30.1
	Test Set: FLORES				
IB+PMI	10.4	14.8	8.1	11.2	10.5
IB+PMI+PIB	13.0	22.0	12.7	15.1	13.8
Samanantar	**30.7**	**36.0**	**30.4**	**28.6**	**27.7**
IB+Samanantar	30.1	35.3	29.1	28.5	26.6

Table 3: Ablation study of the impact of using different fine-tuning corpora sizes (PMI+PIB) and their comparison against a model trained from scratch as well as fine-tuned on a general domain corpus (Samanantar). We evaluate Indic to English translation on the WAT 2021 as well as the FLORES test sets.

reasonable tradeoff given the high levels of compression achieved. Especially given that IndicALBART is 84% smaller than mBART50, means that large capacity GPUs (which not everyone has easy access to) may not be needed. Furthermore, the drops in quality can be addressed via distillation.

Distillation successfully transfers performance from large to smaller models: We see that fine-tuning the pre-trained IndicALBART on distilled data from IndicBART can match the performance of the IndicBART model. Finetuning pre-trained IndicALBART performs better than training a randomly initialized model on the same distilled data in the XX-En direction. On the other hand, both the approaches are competitive in the En-XX direction.

Self-training on distilled data is beneficial: When IndicBART and MB50 are finetuned on distillation data generated from a previously finetuned model, we see significant improvements in the XX-En direction, and modest improvements in the En-XX directions. These observations are mostly in line with Dabre and Fujita (2020).

In summary, compact language family specific pre-trained models are competitive with large universal language models. This can result in reasonable gains in fine-tuning multilingual models (3.3-3.5 hours for IndicBART variants vs 4.7-5 hours for mBART50) and significantly reduce the memory footprint (97-244M vs 611M) for deployment.

4.5 Ablation Studies

We now perform ablation experiments to study the (a.) impact of script unification on translation, (b.) impact of corpora sizes and domains on translation, (c.) translation quality for languages unseen during fine-tuning, and (d.) translation quality on languages unseen during pre-training. Although we train models on all languages, we only report on a subset due to lack of space. Please see Sections C, D in the appendix for more detailed results.

4.5.1 Impact Of Script Unification

Table 2 contains the ablation tests giving the results for the impact of script unification with multilingual fine-tuning. Comparing scores of models fine-tuned on unified script IndicBART (IB+M2O/O2M) against separate script IndicBART (SSIB+M2O/O2M) it is clear that overall, the former is better than the latter which could indicate that script unification enables languages to better benefit from each other. The case of Kannada, Punjabi and Oriya further, illustrates the utility of script unification. The results for these languages are italicized in the rows labelled MB50+Bi and MB50+O2M/M2O in Table 1. mBART50 was not pre-trained on these languages so we converted the training data in these languages in the Devanagari script[7]. With this trick, we still managed to get large performance improvements over the baselines trained from scratch, and these improvements are often close to those exhibited by using IndicBART. This shows that we may not need to pre-train on all languages. However, explicitly training on the languages of interest should lead to better translation quality (Tang et al., 2020b).

4.5.2 Impact Of Corpora Size and Domain

Table 3 shows the impact of corpora sizes as well as training data domain on some Indic to English pairs (complete results in Appendix D). All models are multilingual (M2O), have the same size and are trained on unified script data. In order to clearly assess the impact of domains, we eval-

[7]None of the pre-training languages use the same script as kn, pa, or.

Setting	M2O		O2M	
	kn-en	pa-en	en-kn	en-pa
IB+Full	32.4	35.7	18.5	26.4
IB+Zero	27.5	31.5	6.1	10.4
SSIB+Zero	24.0	28.2	3.9	7.4

Table 4: Evaluation of Kannada and Punjabi to/from English translation, which aren't seen when fine-tuning.

Model	ne-en	si-en
Bi (Scratch)	5.2	4.3
IB+Bi	**10.5**	**8.5**
(Liu et al., 2020)	14.5	13.7

Table 5: Evaluation of Nepali and Sinhala to English translation where IndicBART hasn't seen Nepali and Sinhala during pre-training.

uate on the WAT 2021 as well as the FLORES test sets. Regardless of the test sets or testing domains, comparing rows IB+PMI and IB+PMI+PIB, it is clear that increasing the amount of fine-tuning data has a positive impact on the final translation quality. However, PMI+PIB data is in-domain for the WAT 2021 test set but out-of-domain for the FLORES test set, and the performance on the latter test set still improves.Furthermore, comparing rows IB+PMI+PIB and Samanantar, we can see widely different results depending on the test set. For the WAT 2021 test set, fine-tuning on the PMI+PIB dataset is comparable to training on Samanantar from scratch indicating that for domain specific models, having a small in-domain fine-tuning data is sufficient. On the other hand, on the more general domain FLORES test sets training on the more diverse Samanantar data is clearly better. Finally, the scores in the row IB+Samanantar show that pre-training has minimal impact when the parallel corpora is large, an observation in line with Liu et al. (2020).

4.5.3 Unseen Languages During Fine-Tuning

We evaluate Kannada and Punjabi to/from English translation where the IndicBART model, with and without script unification, is fine-tuned on the multilingual PMI data where the training data for these languages is missing (denoted by "Zero"). We compare against a setting where the training data is used (denoted by "Full"). Table 4 shows what happens when languages are seen during pre-training but not during fine-tuning. There are two critical observations: First, despite not having seen any training data for the given language pairs, we still obtain a reasonable translation for translation into English. However, the quality of translation from English is poor due to the decoder not having seen those specific Indic languages during fine-tuning. Incorporating a monolingual de-noising objective for unseen target languages during finetuning could alleviate this problem. Second, script unification has a large impact on the final performance, often improving performance by up to 3.5 BLEU over a separate script model.

4.5.4 Unseen Languages During Pre-Training

We study Nepalese (ne) and Sinhala (si) to English translation using the parallel training data from Guzmán et al. (2019) (also used in Liu et al. (2020)) for bilingual fine-tuning, and evaluate on the FLORES devtest set[8]. Note that for Sinhala we have to resort to script mapping into Devanagari. Table 5 shows what happens when we perform fine-tuning for languages that IndicBART is not trained on. The baselines, trained using the unified script IndicBART vocabulary, will seem weaker than what is reported in previous work, but it should be noted that the vocabulary was not actually trained for Nepali and Sinhala. Regardless, fine-tuning leads to substantial improvements in translation quality, which indicates the utility of IndicBART even for unseen languages. Comparing against Liu et al. (2020) who use the same fine-tuning data as us but their mBART model is pre-trained on both languages, we can see that our models are not too far behind.

5 Experiments: Extreme Summarization

We compare the performance of fine-tuning IndicBART, its variants and mBART50 on the challenging *extreme summarization* task (Narayan et al., 2018) for Indic languages. The small datasets, enable a good study of the utility of pre-training.

5.1 Models Trained

We fine-tune and compare the mBART50 (MB), IndicBART (IB), IndicALBART (IALB) and the separate script IndicBART model (SSIB) models. Punjabi is not present in mBART50 and has its script mapped to Devanagari before fine-tuning (italicized results).

[8] https://github.com/facebookresearch/flores

Lang	MB50	IB	SSIB	IALB
bn	**21.87**	21.46	20.52	19.86
gu	**18.28**	18.20	16.38	16.81
hi	**31.71**	30.94	30.33	30.04
mr	18.33	**19.00**	18.66	18.44
pa	*22.14*	24.82	**25.08**	23.29
ta	19.50	**20.40**	20.23	17.41
te	13.34	**14.38**	13.34	13.55

Table 6: Rouge-L scores for summarization on XL-Sum.

5.2 Datasets and Preprocessing

We used the multilingual XL-Sum dataset (Hasan et al., 2021) for our experiments. The Indic languages we focus on for evaluating our IndicBART models are: Bengali, Gujarati, Hindi, Marathi, Punjabi, Tamil and Telugu. We use the updated splits of Hasan et al. (2021), the statistics of which are given in their github page[9]. Since the splits are not n-way parallel, we do not conduct multilingual fine-tuning due to potential content overlaps between splits across languages. Like we did in NMT, we map all scripts to Devanagari as applicable for fine-tuning (only Punjabi for mBART50, all languages for IndicBART and IndicALBART and none for separate script IndicBART). Statistics are given in Table 10 in the appendix.

5.3 Model Training Settings

Similar to NMT, we use YANMTT for fine-tuning. We use maximum document-summary lengths of 512-64 tokens which loosely follows previous work (Lewis et al., 2020). Most of the optimal hyperparameters were the same as for NMT. We train our models till convergence on the development set Rouge-L F1 scores (RL) (Lin, 2004). For decoding test sets, we use beam size of 5, length penalty of 1.2 and a decoding n-gram repetition limit of 4[10]. We report RL scores on the decoded results computed using multilingual Rouge scoring toolkit[11]. Refer to section F in the appendix for details.

5.4 Results

Table 6 contains the results for the summarization experiments. IndicBART (IB) and mBART50 are competitive with each other where the former performs slightly better for Marathi, Punjabi, Tamil and Telugu. Once again, separate script IndicBART (SSIB) fared poorer than IndicBART except for Punjabi indicating the importance of script unification. Similar to NMT, fine-tuning IndicALBART gives poorer results often lagging 1-3 RL points behind IndicBART which we consider to be a reasonable tradeoff given the reduced parameter sizes. We expect that distillation may help improve performance, like it does for NMT. Overall, the major conclusions are in line with the those observed for the low-resource NMT task.

6 Conclusion and Future Work

We presented IndicBART, a multilingual, pre-trained sequence-to-sequence model to support development of NLG applications for Indian languages. IndicBART supports 11 Indian languages and English, and utilizes the orthographic similarity of Indic scripts to enable better cross-lingual transfer. IndicBART presents a case-study for language family-specific pre-trained S2S models. Our experiments on fine-tuning IndicBART for NMT and summarization showed that the model is competitive with large models such as mBART50. We further compressed IndicBART while maintaining dowstream task performance via parameter sharing (IndicALBART) combined with multilingual distillation. We showed that script unification has a strong positive impact on translation and summarization. We also showed that IndicBART, thanks to its script independent nature, can be readily used for enabling translation for languages such as Sinhala and Nepali which IndicBART has not been explicitly pre-trained for. Furthermore, we showed that fine-tuning IndicBART on one set of languages enables translation for another unseen set of languages, which shows that pre-trained models enable translation without parallel corpora.

In the future, we plan to support more Indic languages in IndicBART; starting with all the 22[12] languages listed in the 8^{th} schedule of the Indian constitution. Increased language coverage and models with lower compute demands can democratize access to NLP technologies. We also plan to focus on training models on longer text chunks (documents) and larger text corpora, incorporating advances in multilingual pre-training, cross-lingual transfer and cross-lingual tasks for Indian languages.

[9] https://github.com/csebuetnlp/xl-sum/

[10] This means that 4-grams wont be repeated in the output.

[11] https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring

[12] https://www.mha.gov.in/sites/default/files/EighthSchedule_19052017.pdf

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

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.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Raj Dabre and Atsushi Fujita. 2019. Recurrent stacking of layers for compact neural machine translation models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6292–6299.

Raj Dabre and Atsushi Fujita. 2020. Combining sequence distillation and transfer learning for efficient low-resource neural machine translation models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 492–502, Online. Association for Computational Linguistics.

Raj Dabre, Anoop Kunchukuttan, Atsushi Fujita, and Eiichiro Sumita. 2018. Nict's participation in wat 2018: Approaches using multilingualism and recurrently stacked layers. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 282–286. The National University (Phillippines).

Raj Dabre and Eiichiro Sumita. 2021. Yanmtt: Yet another neural machine translation toolkit.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. 2021. Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8584–8595, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *CoRR*, abs/2106.03193.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. 2020. Contact Relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20. In *Conference on Machine Translation*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Barry Haddow and Faheem Kirefu. 2020. PMIndia – A Collection of Parallel Corpora of Languages of India. *arxiv 2001.09907*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., 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, Online. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. 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)*, pages 1312–1323, Online. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

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

Anoop Kunchukuttan, Mitesh Khapra, Gurneet Singh, and Pushpak Bhattacharyya. 2018. Leveraging orthographic similarity for multilingual neural transliteration. *Transactions of the Association for Computational Linguistics*, 6:303–316.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, and Sadao Oda, Yusuke Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation*, Bangkok, Thailand. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6908–6915. AAAI Press.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *CoRR*, abs/2104.05596.

Machel Reid, Junjie Hu, Graham Neubig, and Yutaka Matsuo. 2021. AfroMT: Pretraining strategies and reproducible benchmarks for translation of 8 African languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1306–1320, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Shashank Siripragrada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A multilingual parallel corpora collection effort for Indian languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3743–3751, Marseille, France. European Language Resources Association.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020a. Multilingual translation with extensible multilingual pretraining and finetuning.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020b. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Lang	Mono	Parallel (XX-En)			
		LR			HR
	IC	PMI	PIB	Total	Sam
as	1.4M	-	-	-	-
bn	39.9M	23.3K	91.9K	115.2K	8.4M
en	54.3	-	-	-	-
gu	41.1M	41.5K	58.2K	99.8K	3.0M
hi	63.1M	50.3K	266.5K	316.9K	8.4M
kn	53.3M	28.9K	-	28.9K	4.0M
ml	50.2M	26.9K	43.1K	70.0K	5.8M
mr	34.0M	28.9K	114.2K	143.1K	3.2M
or	7.0M	31.9K	94.4K	126.4K	990.4K
pa	29.2M	28.2K	101,092	129.3K	2.4M
ta	31.5	32.6K	115.9K	148.6K	5.1M
te	47.9M	33.3K	44.7K	78.1K	4.7M
Total	450M	326.3K	930.3K	1.2M	46.2M

Table 7: Statistics of monolingual and parallel corpora (#sentences) for pre-training IndicBART and fine-tuning it, respectively.

A Corpora statistics

Table 7 gives the statistics for the monolingual corpora, Indiccorp (IC), and parallel corpora, PMI, PIB and Samanantar (Sam) used in this paper. Indiccorp is used for pre-training IndicBART and the parallel corpora are used for fine-tuning or for training models from scratch. PMI and PIB have similar domains. PMI is used to simulate a realistic low-resource domain specific setting, and PIB is used to simulate a middle-resource domain specific setting. Samanantar is used to simulate a high resource general domain setting.

B NMT Model Training Settings

We use a single GPU for bilingual and 8 GPUs for multilingual models, all of which are Transformers. Multilingual models are trained using the approach in Johnson et al. (2017). Due to the large number of models we train, we did not perform exhaustive hyperparameter tuning. We mainly focused on tuning the learning rates, batch sizes and warmups. We found that high dropouts were surprisingly ineffective, especially for multilingual settings, regardless of training from scratch or fine-tuning. Nevertheless, for fine-tuning IndicBART and its variants, we determined the following optimal hyperparameters: dropouts of 0.1, label smoothing of 0.1, warmup of 16,000 steps, 2048 tokens per batch per GPU, learning rate of 0.001 and weight decay of 0.00001 with the ADAM optimizer for training. For mBART50, we used warmup of 2,500 steps, 512 tokens per batch per GPU, and learning rate of 0.00003.[13] For bilingual and multilingual models trained from scratch on the small PMI and PIB data, we use smaller models with hidden and filter sizes of 512 and 2048, respectively, while keeping all other hyperparameters the same as for IndicBART which we found to be highly effective. As Samanantar data is much larger, we keep its size the same as IndicBART. Except for separate script IndicBART and mBART50, all models use the same vocabulary as IndicBART for consistency.

We train our models till convergence on the development set BLEU scores (Papineni et al., 2002) which are computed via greedy decoding every 1,000 batches. For multilingual models we use the global development set BLEU score, an average of BLEU scores for each language pair. During decoding the test sets, we use beam search with a beam of size 4 and a length penalty of 0.8. We report the BLEU scores on the decoded results computed using sacreBLEU[14] (Post, 2018).

C NMT Results: Impact of Script Unification

Table 8 contains the results of ablation studies for the impact of script unification in bilingual and multilingual settings. Regardless of bilingual or multilingual fine-tuning, it is clear that script unification tends to give better results on average as compared to using separate scripts to represent all languages.

D NMT Results: Effect of Corpora Size and Domain

Table 9 contains the results showing the impact of varying corpora sizes and domain on translation quality. In the main paper, we could not show results for all languages and directions, due to lack of space. There are three key points to note: (a.) fine-tuning using small in-domain corpora (PMI) gives competitive results compared to using a large general domain corpus. (b.) Additional corpora from a related domain (PMI) leads to substantial improvements in translation quality for in- as well as out-of-domain performance indicating that fine-tuning a pre-trained model on a corpus belonging to a different domain (PMI/PIB) is a viable option in case

[13] A small learning rate is needed since we can train on very small batches given the large model size.

[14] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.5.1

Model	bn	gu	hi	kn	ml	mr	or	pa	ta	te
	XX-En									
IB+M2O	**24.8**	**33.9**	**37.2**	**32.4**	**28.5**	**28.5**	**28.8**	**35.7**	**27.3**	29.5
IB^{noSM}+M2O	24.1	33.8	35.5	31.2	27.9	28.0	28.1	**35.7**	26.9	28.4
IB+Bi	23.6	35.5	36.8	31.6	27.9	26.8	28.3	36.3	27.0	**29.9**
IB^{noSM}+Bi	22.3	34.9	36.6	30.8	27.5	26.7	28.0	36.0	26.3	29.7
	En-XX									
IB+O2M	**9.1**	**24.0**	**27.3**	18.5	**6.7**	**16.7**	12.9	**26.4**	**11.6**	**3.7**
IB^{noSM}+O2M	9.3	**24.0**	**27.3**	17.9	6.2	16.4	**16.6**	23.4	11.4	3.0
IB+Bi	8.2	23.6	26.9	17.7	6.0	15.8	11.8	25.1	10.8	3.6
IB^{noSM}+Bi	8.2	22.9	26.6	17.3	5.8	14.6	14.8	22.9	10.5	3.6

Table 8: Ablation studies to study the impact of multilingualism and script unification on downstream performance of IndicBART. Scores are reported on the WAT 2021 test set.

Model	bn	gu	hi	kn	ml	mr	or	pa	ta	te
	Test Set: WAT 2021									
	XX-En									
IB+PMI	24.8	33.9	37.2	32.4	28.5	28.5	28.8	35.7	27.3	29.5
IB+PMI+PIB	**28.9**	38.8	**41.7**	34.6	**33.2**	**32.5**	**33.2**	41.3	**32.0**	33.0
Samanantar	27.9	**39.0**	41.8	**34.8**	32.7	32.0	32.9	**41.4**	31.2	**34.4**
IB+Samanantar	27.1	38.0	41.0	34.1	31.6	31.1	32.3	40.1	30.1	32.4
	En-XX									
IB+PMI	9.1	24.0	27.3	18.5	6.7	16.7	12.9	26.4	11.6	3.7
IB+PMI+PIB	**11.1**	**25.5**	**33.0**	**18.9**	**7.2**	**19.1**	**14.3**	**27.1**	**13.6**	3.6
Samanantar	9.7	24.7	**33.0**	17.5	7.0	18.4	13.3	25.5	12.7	**5.8**
IB+Samanantar	9.4	24.2	**33.0**	17.2	6.5	17.7	13.5	25.6	11.8	5.6
	Test Set: FLORES									
Model	**bn**	**gu**	**hi**	**kn**	**ml**	**mr**	**or**	**pa**	**ta**	**te**
	XX-En									
IB+PMI	10.4	13.2	14.8	11.8	8.1	10.1	11.2	12.9	10.5	10.5
IB+PMI+PIB	13.0	18.4	22.0	13.1	12.7	16.1	15.1	18.5	13.8	16.2
Samanantar	**30.7**	**33.6**	**36.0**	**27.4**	**30.4**	**30.0**	**28.6**	**34.2**	**27.7**	**32.7**
IB+Samanantar	30.1	32.6	35.3	27.2	29.1	29.6	28.5	33.0	26.6	32.1
	En-XX									
IB+PMI	3.5	9.5	14.7	5.6	2.1	6.0	5.3	10.6	5.0	3.1
IB+PMI+PIB	5.4	13.5	22.8	7.5	2.8	9.1	6.4	15.5	6.9	3.5
Samanantar	**17.3**	**22.6**	**31.3**	**16.7**	**14.2**	**14.7**	**10.1**	**21.9**	**14.9**	**20.4**
IB+Samanantar	17.1	21.5	31.2	16.2	13.0	14.2	10.2	21.5	13.7	19.5

Table 9: Ablation study of the impact of using different sizes of fine-tuning corpora (PMI and its combination with PIB) and their comparison against a model trained from scratch as well as fine-tuned on a general domain corpus (Samanantar). We evaluate on the WAT 2021 as well as the FLORES test sets.

training corpus for the target domain (FLORES) is unavailable. Furthermore, going from low-resource to middle resource settings does not diminish the contribution of pre-trained models. (c.) General domain corpora inevitably lead to the best performance, but since training large models on large general domain corpora is more time-consuming, fine-tuning is a more attractive option since pre-training needs to be done only once.

E Corpora statistics for summarization experiments

Table 10 contains statistics of the Indic section of the XL-sum dataset which we use for summa-

Language	Train	Dev	Test
bn	8,102	1,012	1,012
gu	9,119	1,139	1,139
hi	70,778	8,847	8,847
mr	10,903	1,362	1,362
pa	8,215	1,026	1,026
ta	16,222	2,027	2,027
te	10,421	1,302	1,302

Table 10: Statistics of the Indic portion of the multilingual XL-Sum dataset (Hasan et al., 2021) that we used for training our summarization models.

rization experiments. We preprocess languages by mapping their scripts to Devanagari as applicable (all languages for IndicBART and IndicALBART; none for separate script IndicBART; only Punjabi for mBART50).

F Summarization Model Training Settings

Similar to NMT, we use YANMTT for fine-tuning. We use maximum document-summary lengths of 512-64 tokens which loosely follows previous work (Lewis et al., 2020). Unlike NMT, we do not train models from scratch as they would not work given the small data sizes and difficulty of summarization. For IndicBART and its variants, we determined the following optimal hyperparameters: batch sizes of 4,096 tokens, dropouts of 0.1, label smoothing of 0.1, learning rate warmup steps of 4,000, learning rate of 0.001 and weight decay of 0.00001 with the ADAM optimizer. For mBART50 we use sentence level batching with 2 document-summary pairs per batch and learning rate of 0.00001 which we found to be optimal. We train our models till convergence on the development set Rouge scores (Rouge-L F1) (Lin, 2004) for all languages, which are computed via greedy decoding every 1,000 batches. Similar to NMT, we save the best performing checkpoints for each language. During decoding the test sets, we use beam search with a beam of size 5, length penalty of 1.2 and a decoding n-gram repetition limit of 4-grams[15]. We report Rouge scores on the decoded results computed using multilingual Rouge scoring toolkit[16].

[15]This means that 4-grams wont be repeated in the output.

[16]https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring