IITP-MT at WAT2021: Indic-English Multilingual Neural Machine Translation using Romanized Vocabulary

Ramakrishna Appicharla,* Kamal Kumar Gupta,* Asif Ekbal, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Patna, Bihar, India
{appicharla_2021cs01,kamal.pcs17,asif,pb}@iitp.ac.in

Abstract

This paper describes the systems submitted to WAT 2021 MultiIndicMT shared task by IITP-MT team. We submit two multilingual Neural Machine Translation (NMT) systems (Indic-to-English and English-to-Indic). We romanize all Indic data and create subword vocabulary which is shared between all Indic languages. We use back-translation approach to generate synthetic data which is appended to parallel corpus and used to train our models. The models are evaluated using BLEU, RIBES and AMFM scores with Indic-to-English model achieving 40.08 BLEU for Hindi-English pair and English-to-Indic model achieving 34.48 BLEU for English-Hindi pair. However, we observe that the shared romanized subword vocabulary is not helping English-to-Indic model at the time of generation, leading it to produce poor quality translations for Tamil, Telugu and Malayalam to English pairs with BLEU score of 8.51, 6.25 and 3.79 respectively.

1 Introduction

In this paper, we describe our submission to the MultiIndicMT shared task at the 8th Workshop on Asian Translation [1] (WAT 2021) (Nakazawa et al., 2021). The objective of this shared task is to build Machine Translation (MT) models between 10 Indic languages (Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil, Telugu) and English. We submit two Multilingual Neural Machine Translation models (MNMT): one for XX $\rightarrow$ EN and one for EN $\rightarrow$ XX (here XX denotes a set of all 10 Indic languages).

Multilingual Machine Translation (Dong et al., 2015; Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019; Freitag and Firat, 2020) has gained popularity in recent times due to the ability to train a single model which is capable of translating between multiple language pairs. The main benefit of multilingual model is transfer learning. When a low resource language pair is trained together with a high resource pair, the translation quality of a low resource pair may improve (Zoph et al., 2016; Nguyen and Chiang, 2017). This method of training is more suitable for Indic languages as they are similar to each other (Dabre et al., 2017, 2020) and relatively under-resourced when compared with European languages (Sen et al., 2018).

Romanization is the process of converting characters that are written in various scripts into Latin script. Amrhein and Sennrich (2020) showed that in a transfer learning setting, romanization improves the transfer between related languages that use different scripts. We train two MNMT models, which translate between Indic languages and English with all Indic data romanized. The models are evaluated using the BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and AMFM (Banchs et al., 2015) metrics.

The paper is organized as follows. In section 2, we briefly mention some notable works on multilingual NMT and romanized NMT. In section 3, we describe the systems submitted along with preprocessing and romanization of Indic data. Results are described in section 4. Finally, the work is concluded in section 5.

2 Related Works

Multilingual Machine Translation enabled the ability to deploy a single model for multiple language pairs without training multiple models. Dong et al. (2015) proposes a multi-task learning framework to translate one source language into multiple target languages by adding language specific decoders. Their method has shown improvements over base-

*Equal contribution
[1]Our Team ID: IITP-MT

Proceedings of the 8th Workshop on Asian Translation, pages 238–243
Bangkok, Thailand (online), August 5-6, 2021.

line models which are trained for individual language pairs. Firat et al. (2016) proposes a many-to-many model for multi-way, multilingual translation using shared attention and language specific encoders and decoders. However, with this setting, model parameters will increase as the number of languages increases.

Johnson et al. (2017) use shared encoder-decoder model in which multiple languages share both encoder and decoder also the attention module. This is achieved by combining multiple language pairs data into a single corpus and adding a language tag to every source sentence to specify its target language. This method enables the zero-shot translation, in which the model can generate sentences belonging to a language pair that is not seen at training time. Aharoni et al. (2019) show that multilingual NMT models are capable of handling large number of language pairs. Freitag and Firat (2020) proposes that the use of multi-way alignment information will improve the translation quality of language pairs for which training data is scarce in multilingual settings.

Improving the quality of NMT models with monolingual data is a common approach nowadays, especially in low resource settings. Back-translation Sennrich et al. (2016) is an effective approach to make use of target monolingual data. In this approach, with the help of existing target-to-source MT system target is translated into source and resulting synthetic parallel corpus is combined with clean corpus and used to train source-to-target NMT system. Multi-task learning framework (Zhang and Zong, 2016; Domhan and Hieber, 2017) is another way to utilize monolingual data to improve the performance of NMT.

Recent studies (Du and Way, 2017; Gheini and May, 2019; Briakou and Carpuat, 2019) show that the romanization will improve the performance of NMT system. However these approaches apply romanization at source side only. Amrhein and Sennrich (2020) showed that romanization can be applied on the target side also followed by an additional, learned deromanization step.

In this work, we follow Johnson et al. (2017) method to train multilingual NMT models. We romanize Indic data and use it to train our models. We also follow back-translation approach (Sennrich et al., 2016) to create synthetic parallel data. We report the results of the models which are trained on combined synthetic and clean parallel corpus.

3 System Description

This section describes datasets, preprocessing and experimental setup of our models.

3.1 Datasets

We use MultiIndicMT parallel corpus [2] consisting of following languages: Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Tamil, Telugu and English. It contains the parallel corpora for 10 Indic languages which are translated into English. We also use PMI monolingual corpus [3] to generate synthetic data with back-translation (Sennrich et al., 2016) approach. Table 1 shows the data sizes of corpora used in the experiments. Development and Test sets contain 1,000 and 2,390 sentences respectively for each language pair.

Language	Parallel	Monolingual
Bengali (BN)	1,341,284	117,757
Gujarati (GU)	518,015	125,647
Hindi (HI)	3,069,725	156,605
Kannada (KN)	396,865	79,433
Malayalam (ML)	1,142,053	82,026
Marathi (MR)	621,481	120,362
Odia (OR)	252,160	103,876
Punjabi (PA)	518,508	90,916
Tamil (TA)	1,354,247	91,324
Telugu (TE)	457,453	111,749
English (EN)	-	109,480

Table 1: Language wise training set sizes in terms of number of sentences. **Parallel**: Parallel corpus size of Indic-EN language pair. **Monolingual**: PMI monolingual corpora sizes of all languages.

3.2 Preprocessing and Romanization

We use a Python based transliteration tool [4] to romanize all Indic language data. This tool supports all Indic language scripts that are used in the experiments. It also has deromanization support which maps Latin script into various Indic scripts. We romanize all Indic language data (Amrhein and Sennrich, 2020) (both parallel and monolingual corpora are romanized) and merge all parallel corpora into single corpus. This combined parallel corpus used to train baseline models.

[2] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/indic_wat_2021.tar.gz

[3] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/filteredmono.tar.gz

[4] https://github.com/sanskrit-coders/indic_transliteration

We follow back-translation (Sennrich et al., 2016) approach to generate synthetic parallel corpora. We merge monolingual corpora of all Indic languages and generate synthetic English data using baseline XX → EN model. The resulting synthetic English - Clean Indic parallel corpus is merged with clean English-Indic parallel corpus and used to further train baseline EN → XX model. We also generate synthetic Indic languages data using monolingual English data. We duplicate the monolingual English data 10 times and the baseline EN → XX model is used to generate synthetic Indic data. The reason to duplicate English data is to get equal size synthetic parallel corpus for all Indic languages. The resulting synthetic Indic - Clean English parallel corpus is merged with clean Indic-English parallel corpus and used to further train baseline XX → EN model.

For the training of EN → XX model, we add language tag to start of every source sentence (Johnson et al., 2017) to denote to which language [5] the source should be translated to. We do not use language tags for XX → EN model as the target is English always. All the training data is shuffled before feeding to the models. The training corpus statistics are shown in Table 2. The combined Development set contains 10,000 sentences and is the same for all models. Table 3 shows the contribution of each language pair in the combined training corpus. Hindi-English pair being the most contributing pair with almost 30% and Odia-English pair being least contributing pair with 3.3%, in both directions.

Model	Train
XX → EN	9,671,791
XX → EN + BT	10,766,591
EN → XX	9,671,791
EN → XX + BT	10,751,486

Table 2: Training data sizes of combined corpora. {XX, EN} → {EN, XX} denotes training data sizes of Baseline models. BT denotes total training data sizes after adding synthetic back-translated parallel corpora.

3.3 Experimental Setup

We train two multilingual models namely XX → EN (Indic languages to English) and EN → XX (English to Indic languages). All the models are trained on the Transformer architecture (Vaswani et al., 2017). We use 6 layered Encoder-Decoder stacks with 8 attention heads. Embedding size and hidden sizes are set to 512, dropout rate is set to 0.1. Feed-forward layer consists of 2048 cells. Adam optimizer (Kingma and Ba, 2015) is used for training with 8,000 warm up steps with initial learning rate of 2. We split the training data of baseline models into subwords with the unigram language model (Kudo, 2018) using SentencePiece (Kudo and Richardson, 2018) implementation. We create two subword vocabularies, one for English and one for all romanized Indic data [6]. The size of English subword vocabulary is 60K and of Indic languages is 100K, for both the models. We use OpenNMT toolkit (Klein et al., 2017)[7] to train our models with batch size of 2048 tokens. Models are evaluated on development sets after every 10,000 steps and checkpoints are created. The baseline models are trained for 100,000 steps and the last checkpoint is used to create a synthetic corpus with the back-translation approach as described in Section 3.2. After creating synthetic parallel corpora, baseline models are further trained for another 200,000 steps [8] on combined synthetic and clean parallel corpora (see Table 2). Finally, all checkpoints that are created by the model using the combined corpora are averaged [9] and considered as the best parameters for each model and used to test our models. We

Language Pair	XX → EN	EN → XX
HI-EN	29.53	30.0
TA-EN	13.60	13.45
BN-EN	13.47	13.57
ML-EN	11.62	11.38
MR-EN	6.79	6.90
GU-EN	5.83	6.0
PA-EN	5.83	5.67
TE-EN	5.27	5.29
KN-EN	4.70	4.43
OR-EN	3.36	3.31

Table 3: Contribution of each language pair (in %) in the training set after merging clean corpus with synthetic back-translated corpus. **XX → EN**: Indic-to-English model. **EN → XX**: English-to-Indic model.

[5] We use following tags: ##2BN, ##2GU, ##2HI, ##2KN, ##2ML, ##2MR, ##2OR, ##2PA, ##2TA, ##2TE

[6] All Indic languages data is merged after romanization and created subword vocabulary on combined corpus.

[7] https://github.com/OpenNMT/OpenNMT-py/tree/1.2.0

[8] We stop the training as there is no improvement in terms of perplexity of models on training data.

[9] OpenNMT-py provides script to average model weights.

Language Pair	XX → EN			EN → XX		
	BLEU	RIBES	AMFM	BLEU	RIBES	AMFM
BN-EN	25.77	0.77	0.78	11.04	0.70	0.73
GU-EN	36.49	0.83	0.81	20.46	0.75	0.81
HI-EN	40.08	0.85	0.83	34.48	0.84	0.82
KN-EN	31.24	0.81	0.80	13.22	0.64	0.79
ML-EN	29.37	0.80	0.80	3.79	0.44	0.76
MR-EN	29.96	0.80	0.80	13.95	0.67	0.80
OR-EN	31.19	0.79	0.80	12.57	0.71	0.74
PA-EN	38.41	0.84	0.82	16.81	0.79	0.66
TA-EN	27.76	0.79	0.79	8.51	0.58	0.76
TE-EN	28.13	0.78	0.78	6.25	0.53	0.76

Table 4: Official BLEU, RIBES and AMFM scores of multilingual models for each language pair. **XX → EN** denotes score of Indic-to-English model. **EN → XX** denotes score of English-to-Indic model.

keep OpenNMT-py's default beam size of 5 during back-translation and inference. For the EN → XX model, after getting the model predictions on the test set, we deromanize these predictions and convert them into respective language scripts.

4 Results and Analysis

The official BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and AMFM (Banchs et al., 2015) scores of the multilingual models are shown in Table 4. We observe that the XX → EN model performance is consistent across all language pairs in terms of all the three scores. HI-EN being the most contributing pair (see Table 3), achieves the BLEU score of 40.08 points. Even the language pair with the least amount of data (OR-EN) yield a BLEU score of 31.19 points. However, we do not observe the same with EN → XX model. The performance of EN → XX model is inconsistent with achieving a high BLEU score of 34.48 points (EN-HI) and least BLEU score of 3.79 (ML-EN). We observe same in terms of RIBES score also. However, AMFM scores of EN → XX model are quite consistent despite having less BLEU and RIBES scores for some language pairs.

Sen et al. (2018) observe that, in the multilingual setting where a single decoder has to handle information about more languages (7 in their case), the performance of the model is limited because of different vocabulary and different linguistic features. In our case, we romanize all data and feed it to the model. Still the EN → XX model is unable to produce good quality translations. We believe that the main reason for such low quality translations is the romanized subword vocabulary, which is shared across 10 different languages, is not helping decoder at the time of generation. There can be two possible ways to fix this issue. One is, using a larger target vocabulary size as 100K subword vocabulary is not giving good results in our case. Another is, creating separate vocabularies for each language instead of combining them together and creating a joint vocabulary, while the data being romanized.

5 Conclusion

In this paper, we describe our submission to the MultiIndicMT shared task to WAT 2021. We submit two multilingual NMT models: many-to-one (10 Indic languages to English) and one-to-many (English to 10 Indic languages). We romanize all Indic language data to convert all languages' tokens in roman script. We also generate synthetic data using the back-translation approach. We train our models on the romanized data sets which is a combination of clean corpora and synthetic back-translated corpora. We evaluate our models using BLEU, RIBES and AMFM scores and observed that many-to-one model achieves highest BLEU score of 40.08 for Hindi-English pair and one-to-many model achieves highest BLEU score of 34.48 for English-Hindi pair. However, the shared subword vocabulary at target side for the one-to-many model lead to the poor performance of the one-to-many model especially in Tamil, Telugu and Malayalam to English pairs by achieving BLEU score of 8.51, 6.25 and 3.79 respectively.

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.

Chantal Amrhein and Rico Sennrich. 2020. On Romanization for model transfer between scripts in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2461–2469, Online. Association for Computational Linguistics.

Rafael E. Banchs, Luis F. D'Haro, and Haizhou Li. 2015. Adequacy–fluency metrics: Evaluating mt in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):472–482.

Eleftheria Briakou and Marine Carpuat. 2019. The University of Maryland's Kazakh-English neural machine translation system at WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 134–140, Florence, Italy. 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, Fabien Cromierès, and Sadao Kurohashi. 2017. Enabling multi-source neural machine translation by concatenating source sentences in multiple languages. *CoRR*, abs/1702.06135.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Jinhua Du and Andy Way. 2017. Pinyin as subword unit for chinese-sourced neural machine translation. In *Proceedings of the 25th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Ireland, December 7 - 8, 2017, volume 2086 of CEUR Workshop Proceedings*, page 89–101. CEUR-WS.org.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

Mozhdeh Gheini and Jonathan May. 2019. A universal parent model for low-resource neural machine translation transfer. *arXiv preprint arXiv:1909.06516*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. 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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. 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.

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.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2018. IITP-MT at WAT2018: Transformer-based multilingual indic-English neural machine translation system. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.