IOL Research Machine Translation Systems for WMT23 Low-Resource Indic Language Translation Shared Task

Wenbo Zhang, Zeyu Yan, Qiaobo Deng, Jie Cai, and **Hongbao Mao**
Transn IOL Research, Wuhan, China

Abstract

This paper describes the IOL Research team's submission systems for the WMT23 low-resource Indic language translation shared task. We participated in 4 language pairs, including en↔as, en↔mz, en↔kha, en↔mn. We use transformer based neural network architecture to train our machine translation models. Overall, the core of our system is to improve the quality of low resource translation by utilizing monolingual data through pre-training and data augmentation. We first trained two denoising language models similar to T5 and BART using monolingual data, and then used parallel data to fine-tune the pretrained language models to obtain two multilingual machine translation models. The multilingual machine translation models can be used to translate English monolingual data into other multilingual data, forming multilingual parallel data as augmented data. We trained multiple translation models from scratch using augmented data and real parallel data to build the final submission systems by model ensemble. Experimental results show that our method greatly improves the BLEU scores for translation of these four language pairs.

1 Introduction

This paper describes our submissions for the WMT23 low-resource Indic language translation shared task. We participated in 4 language pairs, including English↔Assamese (en↔as), English↔Mizo (en↔mz), English↔Khasi (en↔kha), and English↔Manipuri (en↔mn).

Our core approach is based on denoising language model pre-training(Devlin et al., 2019; Lample and Conneau, 2019; Song et al., 2019; Raffel et al., 2019; Lewis et al., 2020) and back-translation(Sennrich et al., 2016a) based data augmentation. Neural machine translation methods are almost the first choice for implementing translation systems at present, but they have certain requirements on the amount of parallel corpora. Low-resource or even zero-resource neural machine translation has been a daunting challenge due to the lack of adequate parallel corpora. Pre-training methods are popular solutions for low-resource cases. When the model parameter scale is large enough and there is enough training data, this method can even perform well in zero resource situations. For the machine translation task, as early as around 2019, XLM(Lample and Conneau, 2019) and MASS(Song et al., 2019) were able to build unsupervised machine translation systems with near-supervised effects using only monolingual data. Now, more advanced pre-training methods like BART(Lewis et al., 2020) and T5(Raffel et al., 2019) are popular choices for training machine translation models in low-resource situations. Therefore, in this paper, referring to the training methods of BART and T5, we trained a T5-style pre-training model and a BART-style pre-training model from scratch using monolingual data. Back-translation is a commonly used method in the field of machine translation. Whether it is low-resource, medium-resource or high-resource, this approach can almost help the model to obtain further improvements on the original basis. Therefore, we also use back-translation to help us further improve the translation quality.

The layout of the subsequent paper is as follows: In Section 2 We introduce the data source and processing strategy; In Section 3 we describes the implementation process of our translation systems; In Section 4 we describe the experimental settings and results; Finally, the conclusion is drawn in Section 5.

2 Data

2.1 Data Source

Bilingual corpus We just used the official en↔as, en↔mz, en↔kha, and en↔mn parallel data(Pal

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 978–982
December 6–7, 2023.

Data	en↔as	en↔mz	en↔kha	en↔mn
Bilingual Data	49808	49575	23996	20990

Table 1: Statistics of bilingual data

Data	en	as	mz	kha	mn
Monolingual Data	60598321	2206328	1864322	178036	298072

Table 2: Statistics of monolingual data

et al., 2023).

Monolingual corpus of Indic languages We also used only official monolingual data for Assamese, Mizo, Khasi and Manipuri.

English monolingual corpus Since the official did not provide English monolingual data, we obtained English monolingual data from the WMT23 general task. Specifically, we used the English side of bilingual data (English↔German and English↔Japanese) in the WMT23 general task as English monolingual data.

2.2 Data Preprocessing

For English monolingual data, we first filter out noisy sentences according to following rules:

- Remove invisible characters.
- Remove sentences containing too more than 300 words or more than 1000 characters or less than 3 characters.
- Remove English sentences containing words exceeding than 40 characters.
- Remove sentences that contain too many punctuation marks.
- Remove sentences that contain repeated substrings, which refers to a string composed of a single character that repeats more than 10 times, or two or more character that repeat more than 5 times.
- Remove sentences that contain HTML tags.
- Convert full-width characters to half-width characters.
- Remove duplicated sentence pairs.

Since all the officially provided data have been tokenized, we used the Moses scripts[1] to do tokenization for English monolingual data. Then we use an n-gram language model trained with KenLM(Heafield, 2011)[2] to calculate the perplexity of English monolingual data and remove sentences with high perplexity(more than 10 000). We just did deduplication for the official data, because the size of the official data is relatively small and the quality is high enough. The amount of data after processing is shown in Table 1 and 2.

We used the Sentencepiece(Kudo and Richardson, 2018) tool to train a multilingual BPE(Sennrich et al., 2016b) model for subword segmentation. Its training data includes all official training data and 2.5 million random samples from English monolingual data. The vocabulary size is set to 48 000.

3 System Overview

We chose Transformer(Vaswani et al., 2017) with pre-norm as our base translation model. In general, our procedure for improving the quality of low-resource translations is divided into two phases, an improvement phase based on pre-training methods and an improvement phase based on data augmentation. Instead of using the pre-trained model to initialize the parameters of the translation model, the pre-training phase merely provides synthetic data for the data augmentation phase, which means that the translation model in the data augmentation phase is trained from scratch. In addition to this, we also used model ensemble in the final submissions.

3.1 Pre-training

The pre-training phase is divided into two steps. In the first step, pre-training for the denoising autoencoder tasks are performed using monolingual data. In the second step, the pre-trained models are fine-tuned using bilingual data. We trained two denoising pre-training models, namely the T5-style(Raffel et al., 2019) model and the BART-style(Lewis et al., 2020) model. The training details

[1]https://github.com/moses-smt/mosesdecoder/

[2]https://github.com/kpu/kenlm

Original sentence	Since their articles appeared , the price of gold has moved up still further .
T5-style input sentence	Since their articles appeared , gold has moved up still further
T5-style target sentence	the price of .
BART-style input sentence	Since their of gold has up still **moved** further .
BART-style target sentence	Since their articles appeared , the price of gold has **moved** up still further .

Table 3: Examples of T5-style and BART-style training data

of the two models are as follows.

As shown in Table 3, Both T5-style and BART-style models are trained by recovering original sentences from corrupted sentences, which are produced by randomly replacing some fragments in the sentences with the mark. The most important difference is that the T5-style target sentence, that is, the label contains only the replaced part, while the BART-style label is the entire original sentence. Another difference is that in this paper we also randomly swap the two words that are not masked in BART-style input sentences. For both models, the proportion of replaced words is 0.15, and the length of replaced segments is 3. We randomly swap words in BART-style input sentences with a probability of 0.5.

We used monolingual data containing 5 languages to train the pre-training models, and then fine-tuned the pre-trained models using parallel corpora containing 4 language pairs in 8 translation directions. In order to keep the number of all languages balanced, we only used 3 million additional English monolingual data at this phase.

3.2 Data Augmentation

The pre-training phase is also divided into two steps, pre-training on synthetic data and fine-tuning on the real bilingual data. We employed the approach inspired by the back-translation(Sennrich et al., 2016a) and Zan et al. (2022) to generate synthetic data. Since we planed to train a multilingual translation model, in order to share knowledge across multiple languages, the synthetic data we generated contains 5 languages and 20 translation directions. In detail, by beam search, we translated an English monolingual sentence into 4 other languages, where any two sentences in different languages are also aligned as they are both translated from the same English sentence. To ensure the quality of the synthesized data, we also calculated the translation perplexity score from Indic languages to English direction via a multilingual translation model from pre-training phase and removed sentence pairs with high perplexity scores. For data diversity, we used both T5-style and BART-style pre-trained models to generate synthetic data, and leveraged the other model to compute the perplexity score, for example, the data generated by the T5-style pre-trained model is scored using the BART-style pre-trained model.

3.3 Model Ensemble

A well-known model ensemble trick is to increase the diversity between different models. However, we did not train multiple translation models from scratch due to time and computational resource constraints. Instead, we fine-tuned the three models, many-to-many, one-to-many, and many-to-one, based on model trained on synthetic data, and then selected the many-to-many and one-to-many or many-to-one models to complete the final submission by model ensemble.

4 Experiments

4.1 Experiment Settings

All of our translation models were implemented based on fairseq(Ott et al., 2019) and trained on 8 NVIDIA A100 GPUs. During training, we used the Adam(Kingma and Ba, 2014) optimizer with $\beta 1 = 0.9, \beta 2 = 0.98$, the learning rate scheduling strategy of inverse sqrt, the number of warmup step set to 4000, the maximum learning rate set to 0.0005 and FP16 to accelerate the training process.

We trained three models, Many2Many(M2M), One2Many(O2M), Many2One(M2O), with 12-encoder, 12-decoder transformer-big model as baselines. They were trained only on a real parallel corpus, with a batch size set at 13,000 tokens. For the models in the Pre-training phase, we used the same model structure as the baselines but with a batch size of 1 million. For the models in the data augmentation phase, we changed the number of layers of models to 10, and the embedding size to 1536.

System	en→as	en→kha	en→mni	en→mz	as→en	kha→en	mni→en	mz→en
O2M Baseline	5.1	11.8	9.1	15.0	-	-	-	-
M2O Baseline	-	-	-	-	14.3	10.6	19.8	18.8
M2M Baseline	7.0	14.8	13.4	19.2	15.9	11.7	23.3	20.6
BART-style Pre-training	11.4	19.3	20.1	25.2	22.4	15.1	35.4	26.5
T5-style Pre-training	12.0	19.6	21.5	26.3	23.6	16.4	35.6	26.9
O2M Data Augmentation	13.0	21.3	23.3	27.4	-	-	-	-
M2O Data Augmentation	-	-	-	-	28.2	20.1	42.1	31.8
M2M Data Augmentation	12.8	21.0	23.4	27.3	25.2	18.0	40.6	29.1
Model Ensemble	13.4	21.6	23.9	27.8	28.6	20.8	42.9	32.4

Table 4: BLEU scores of all translation direction on validation sets

4.2 Results

All experiments were evaluated using the sacrebleu(Post, 2018) tool to calculate BLEU(Papineni et al., 2002) scores on the official validation sets, and we did not detok before calculating the BLEU scores. We used beam search with beam size=5 to decode all models and the results are shown in Table 4.

According to Table 4, it can be seen that the many-to-many baseline preforms better than one-to-many and many-to-one. I believe this is because the parallel corpus size is too small where the many-to-many model can share knowledge across different languages. Both BART-style and T5-style pre-training significantly improved BLEU scores in all directions, with T5-style slightly better than BART-style. All translation directions are further improved after data augmentation. When English is the source language, the improvement is small, and when English is the target language, the improvement is larger. This is because this phase is mainly based on a large amount of real English monolingual data. The one-to-many and many-to-one models perform equally or better than the many-to-many model at this phrase, as there is no longer a severe lack of linguistic knowledge. Finally, the model ensemble helps the system to obtain further improvements.

5 Conclusion

In this paper, we describe IOL Research's submission to the WMT2023 low-resource Indic language translation shared task. We participated in four sub-tasks with a total of eight translation directions. Our system mainly improves the translation quality of these languages in the low-resource case through pre-training and data augmentation. Experimental results show that we achieved large improvements in all directions.

References

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.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

DiederikP. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv: Learning,arXiv: Learning*.

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*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv: Computation and Language,arXiv: Computation and Language*.

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*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Santanu Pal, Partha Pakray, Sahinur Rahman Laskar, Sandeep Kumar Dash, Lenin Laitonjam, Vanlalmuansangi Khenglawt, Sunita Warjri, and Pankaj Kundan Dadure. 2023. Findings of the wmt 2023 shared task on low-resource indic language translation. In *Proceedings of the Eighth Conference on Machine Translation*, Singapore. Association for Computational Linguistics.

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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and PeterJ. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv: Learning,arXiv: Learning*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv: Computation and Language,arXiv: Computation and Language*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Changtong Zan, Keqin Peng, Liang Ding, Baopu Qiu, Boan Liu, Shwai He, Qingyu Lu, Zheng Zhang, Chuang Liu, Weifeng Liu, Yibing Zhan, and Dacheng Tao. 2022. Vega-MT: The JD explore academy machine translation system for WMT22. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 411–422, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.