Iterative Multilingual Neural Machine Translation for Less-Common and Zero-Resource Language Pairs

Minh-Thuan Nguyen, Phuong-Thai Nguyen, Van-Vinh Nguyen, Minh-Cong Nguyen Hoang
Department of Computer Science, University of Engineering and Technology, VNU Hanoi
{thuannm, thainp, vinhnv}@vnu.edu.vn
minhcongnguyen1508@gmail.com

Abstract

Research on providing machine translation systems for unseen language pairs is gaining increasing attention in recent years. However, the quality of their systems is poor for most language pairs, especially for less-common pairs such as Khmer-Vietnamese. In this paper, we show a simple iterative training-generating-filtering-training process that utilizes all available pivot parallel data to generate synthetic data for unseen directions. In addition, we propose a filtering method based on word alignments and the longest parallel phrase to filter out noise sentence pairs in the synthetic data. Experiment results on zero-shot Khmer→Vietnamese and Indonesian→Vietnamese directions show that our proposed model outperforms some strong baselines and achieves a promising result under the zero-resource condition on ALT benchmarks. Besides, the results also indicate that our model can easily improve their quality with a small amount of real parallel data.

1 Introduction

Neural Machine Translation (NMT) has recently achieved impressive performance on high-resource language pairs which have large amounts of parallel training data (Wu et al., 2016) (Vaswani et al., 2017). However, these systems still work poorly when the parallel data is low or unavailable. Research on zero-resource language pairs is gaining much attention in recent years, and it has been found to use pivot language, zero-shot NMT, or zero-resource NMT approaches to deal with the translation of unseen language pairs.

In pivot language approaches, sentences are first translated from the source language into the pivot language through a source-pivot system, and then from the pivot language into the target language by using a pivot-target system. Although this simple process has shown strong translation performance (Johnson et al., 2017), it has a few limitations. The pivoting translation process at least doubles decoding time during inference because more than one pivot language may be required to translate from the source to the target language. Additionally, translation errors compound in a pipeline.

Zero-shot NMT approaches are inspirited from multilingual NMT (multi-NMT) systems that use only one encoder and one decoder to represent multiple languages in the same vector space, hence it should be possible to take advantage of data from high-resource language pairs to improve the translation of low-resource language pairs. (Ha et al., 2016; Johnson et al., 2017) showed that the zero-shot systems are able to generate reasonable output at the target language by adding the desired output language's language tag at the beginning of the source sentence. Note that there is no direct parallel data between the source and target languages during training. However, the performance of these approaches is still poor when the source and target languages are unrelated or the observed language pairs are not enough to capture the relation of unseen language pairs.

Similar to the above approaches, zero-resource NMT approaches do not use any direct source-target parallel corpus, but the approaches focus on generating pseudo-parallel corpus by using back-translation to translate sentences in the pivot language of the pivot-target parallel corpus to the source language (Lakew et al., 2017; Gu et al., 2019). One of the main limitations of these approaches is that the source between training and testing scenarios are different since the source in training is synthetic. However, the approaches still outperform pivot language and zero-shot NMT approaches because they can potentially utilize all available parallel and monolingual corpus (Currey and Heafield, 2019).

In this work, our main contributions are (1) improving the quality of zero-resource NMT by introducing a simple iterative *training-generating-filtering-training* process and (2) proposing a noise filtering method. Especially, we evaluate our approach on less-common and low-resource language pairs such as Khmer-Vietnamese. In this scenario, source-pivot (Khmer-English) and pivot-target (English-Vietnamese) pairs are also low-resource (pivot is often English). Our approach starts from a multilingual NMT system that is trained on source-pivot and pivot-language pairs, the system then generates source-target synthetic corpus by back-translating the pivot side of the pivot-target corpus to the source language. Next, We filter out poor translations in the generated translations by applying our proposed data filtering method based on word alignments and the longest parallel phrase. After that, the multilingual NMT system is continuously trained on both the filtered synthesis data and the original training data, we repeat this *training-generating-filtering-training* cycle for a few iterations. As a result, our experiments showed that by adding the filtered synthetic corpus, our model outperformed the pivot, zero-shot, and zero-resource baselines over zero-shot Khmer→Vietnamese and Indonesian→Vietnamese directions on the Asian Language Treebank (ALT) Parallel Corpus (Riza et al., 2016). Moreover, the experiment results indicate that our model can easily improve their quality with a small amount of real parallel data.

The rest of this paper is organized as follows. We first review relevant works on translation for zero-resource language pairs in Section 2, then introduce some background and related formulas in Section 3. Next, we show our approach in Section 4. After that, we illustrate our experiments and results in Section 5. Finally, our conclusion is presented in Section 6.

2 Related Work

Training a machine translation system for translating unseen language pairs has received much interest from researchers in recent years. This section discusses relevant works on zero-shot and zero-resource NMT, which are related to our approach.

Zero-shot NMT

(Ha et al., 2016; Johnson et al., 2017) showed that using a single NMT can learn to translate between language pairs it has never seen during training (zero-shot translation). Their solution does not require any changes to the traditional NMT model architecture. Instead, they add an artificial token at the beginning of the source sentence to specify the required target language. Although this approach illustrated promising results for some untrained language pairs such as from Portuguese to Spanish, its performance is often not good enough to be useful and lags behind pivoting. In our work, we use this system as an initial multi-NMT system.

(Arivazhagan et al., 2019) pointed out that the success of zero-shot translation depends on the ability of the model to capture language invariant features for cross-lingual transfer. Therefore, they proposed two classes of auxiliary losses to align the source and pivot vector spaces. The first minimizes the discrepancy between the feature distributions by minimizing a domain adversarial loss (Gani et al., 2015) that trains a discriminator to distinguish between different encoder languages using representations from an adversarial encoder. The second takes advantage of available parallel data to enforce alignment between the source and the pivot language at the instance level. However, this approach does not work for less-common language pairs such as Khmer-Vietnamese since the size of multilingual training data including source-pivot and pivot-target is low, so it is not enough to capture the language invariant features.

Zero-resource NMT

(Lakew et al., 2017) used a multilingual NMT system to generate zero-shot translations on some portion of the training data, then re-start the training process on both the multilingual data and the generated translations. By adding the synthetic corpus, the model can alleviate the spurious correlation problem. This work is similar to our work but they did not filter out noise sentence pairs in the synthetic corpus.

(Currey and Heafield, 2019) augmented zero-resource NMT with monolingual data from the pivot language. The authors pointed out that the pivot language is often high-resource language and more high-quality than the monolingual source or target language (pivot language is often English), so leveraging the monolingual pivot language data is worthwhile to enhance the quality of zero-resource NMT systems.

3 Background

3.1 Neural Machine Translation

The standard NMT architecture contains an encoder, a decoder and an attention-mechanism, which are trained with maximum likelihood in an end-to-end system (Bahdanau et al., 2014). Assume the source sentence and its translation are $x = \{x_1, ..., x_{T_x}\}$ and $y = \{y_1, ..., y_{T_y}\}$ respectively.

Encoder is a bidirectional Recurrent Neural Network (RNN) (Schuster and Paliwal, 1997) that encodes the source sentence into a sequence of hidden state vectors, the hidden state vector of word x_i is $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$, where $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ are forward and backward hidden state respectively.

$$\overrightarrow{h_i} = f(e_{x_i}, \overrightarrow{h}_{i-1}) \quad (1)$$

$$\overleftarrow{h_i} = f(e_{x_i}, \overleftarrow{h}_{i+1}) \quad (2)$$

Note that e_{x_i} is the vector of word x_i, f is a non-linear function such as Long Short-term Memory (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (Cho et al., 2014).

Attention is a mechanism used to compute a context vector by searching through the source sentence at each decoding step (Bahdanau et al., 2014). At the j-th step, the score between the target word y_j and the i-th source word is computed and normalized as below:

$$e_{i,j} = v_a^T \tanh(W_a s_{j-1} + U_a h_i) \quad (3)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{i'=1}^{T_x} \exp(e_{i'j})} \quad (4)$$

The context vector c_j is computed as a weighted sum of all source hidden states:

$$c_j = \sum_{i=1}^{T_x} \alpha_{ij} h_i \quad (5)$$

Decoder is a unidirectional RNN which uses the representation of the encoder and the context vector to predict words in the target language. At the j-th step, the target hidden state s_j is computed by:

$$s_j = f(e_{y_{j-1}}, s_{j-1}, c_j) \quad (6)$$

Given the previous predicted words $y_{<j} = \{y_1, ..., y_{j-1}\}$, the context vector c_j and the target hidden state s_j, the decoder is trained to predict the next word y_j as follows:

$$p(y_j | y_{<j}, s_j, c_j) = \text{softmax}(W_o t_j) \quad (7)$$

$$t_j = g(e_{y_{j-1}}, c_j, s_j) \quad (8)$$

where g is a nonlinear function, W_o is used to output a vocabulary-sized vector.

3.2 Multilingual NMT

(Ha et al., 2016; Johnson et al., 2017) indicated a simple approach to use a standard NMT system to translate between multiple languages. This system leverages the knowledge from translation between multiple languages and is referred to as a multilingual NMT system. In order to make use of multilingual data containing multiple language pairs into the standard NMT system, authors proposed one simple modification to the input data, which is to add an artificial token at the beginning of the input sentence to indicate the desired target language. After adding the token to the input data, over-sampling or under-sampling techniques are applied to balance the ratio of language pairs in the multilingual data, and the model is trained with all the multilingual data at once. Besides, a shared wordpiece model (Sennrich et al., 2015) across all the source and target data is used to address the problem of translation of unknown words and limitation of the vocabulary for computational efficiency, usually with 32,000 word pieces.

4 Approach

This paper concentrates on improving the quality of zero-resource NMT between two languages X and Y given a pivot language Z. We assume that we have $X \leftrightarrow Z$ and $Z \leftrightarrow Y$ parallel data, but no direct $X \leftrightarrow Y$ data. Algorithm 1 represents our proposed training process. Notably, our experiments focus on less-common and low-resource language pairs such as Khmer-Vietnamese, Indonesian-Vietnamese, so the amount of $X \leftrightarrow Z$ and $Z \leftrightarrow Y$ parallel data is quite small. Therefore, in order to build a good initial multi-NMT model, the first step of our work is to augment the multilingual training data that is shown in Section 4.1. Take a look at the Algorithm 1, given an initial training data D including $X \leftrightarrow Y$ and $Y \leftrightarrow Z$ parallel data, our training process contains four main steps which are iterated for multiple times.

Algorithm 1: Iterative Multi-NMT with Data Filtering Procedure

```
1:  $D = (X \leftrightarrow Z, Z \leftrightarrow Y)$
2:  **repeat**
3:    Multi-NMT $\leftarrow$ training using dataset D
4:    **for each** Z in $(Z \leftrightarrow Y)$ **do**
5:      $X^* \leftarrow$ Multi-NMT(Z), generating
6:    **end for**
7:    $S \leftarrow (X^* \leftrightarrow Y)$, synthetic data
8:    $F \leftarrow$ Filter(S), filtering synthetic data
9:    $D \leftarrow D \cup F$
10: **until** Multi-NMT converges
```

Figure 1: Algorithm of the proposed approach using iterative multi-NMT with data filtering.

- Step 1 (line 3): Train a multilingual NMT by using the training dataset D.
- Step 2 (line 4, 5, 6): Generate $(X^* \rightarrow Y)$ synthetic parallel data by using the trained multi-NMT model to translate sentences from pivot language Z in $(Z \leftrightarrow Y)$ to language X. We can obtain more synthetic data $(X \leftrightarrow Y)$ by translating sentences from pivot language Z in $(X \leftrightarrow Z)$ to language Y.
- Step 3 (line 8): Filter the synthetic data to eliminate bad parallel sentence pairs by using data selection techniques (See Section 4.2).
- Step 4 (line 9): Expand the multilingual training data by adding the filtered synthetic data F to the original training data D.

In our *training-generating-filtering-training* cycle, new synthetic $X \leftrightarrow Y$ data is generated at each iteration. We expect that by adding this synthetic data, the multi-NMT model not only improves the translation of zero-shot directions between X and Y but also boosts other directions such as between X and Z, Y and Z. Therefore, round after round, we can build a better multi-NMT system with the synthetic data. Use this better system in order to generate new synthetic data, then use this data with the original training data to build an even better system. Finally, this cycle continues until the model converges.

4.1 Data Augmentation

As mentioned above, if the amount of multilingual training data is too small, the multi-NMT system is unable to learn to translate between zero-shot directions. Hence, in our work, to augment the parallel data for $(X \leftrightarrow Z)$ and $(Z \leftrightarrow Y)$, we leverage monolingual data in both target and source side by using back-translation (Sennrich et al., 2016) and self-training (Zhang and Zong, 2016). Given a parallel data $(X \leftrightarrow Z)$ and monolingual data M_X, M_Z in language X, Z respectively, we denote by $\overrightarrow{f}$ and $\overleftarrow{g}$ the forward (from X to Z) and the backward (from Z to X) NMT systems.

Back-translation is a popular data augmentation method utilizing target side monolingual data. To perform back-translation, given the parallel data $(X \leftrightarrow Z)$, a base backward NMT system $\overleftarrow{g}$ is trained and use it to translate M_Z to language X, denoted by $\overleftarrow{g}(M_Z)$. The original parallel data $(X \leftrightarrow Z)$ is then concatenated with the back-translated data $(\overleftarrow{g}(M_Z) \leftrightarrow M_Z)$ to obtain a new training data.

Self-Training augments the original training data by first training a base forward NMT system $\overrightarrow{f}$ on $(X \leftrightarrow Z)$ data, then use this trained model to translate M_X to language Z, denoted by $\overrightarrow{f}(M_X)$. The new synthetic data $(M_X \leftrightarrow \overrightarrow{f}(M_X))$ is also combined with the original training data to obtain a new training dataset.

In our work, we augment parallel data by using both these two methods because they are complementary to each other. The original training data is combined with back-translated and self-trained data to obtained the augmented parallel data, $(X \leftrightarrow Z) \cup (\overleftarrow{g}(M_Z) \leftrightarrow M_Z) \cup (M_X \leftrightarrow \overrightarrow{f}(M_X)$.

4.2 Data Filtering

Combining synthetic data with the multilingual training data is a simple and effective way to boost the quality of zero-shot directions in zero-shot NMT and zero-resource NMT systems (Lakew et al., 2017; Currey and Heafield, 2019). However, the synthetic data potentially contains a lot of noise—translation errors, since it is often generated by using back-translation or self-training. Therefore, in this section, we show our proposed method to filter noise sentence pairs from synthetic data based on sentence semantic similarity. As described in Section 4, a synthetic sentence pair (x_i, y_i) is generated by translating z_i in $(Z \leftrightarrow Y)$ data to x_i. We consider that (x_i, y_i) is good synthetic sentence pair if x_i is both semantically similar to y_i and z_i. A semantic score for each synthetic sentence x_i is computed as below:

$$\text{score}(x_i) = \frac{\text{sim}(x_i, y_i) + \text{sim}(x_i, z_i)}{2} \tag{9}$$

where $sim(x_i, y_i)$ and $sim(x_i, z_i)$ are the semantic similarity of (x_i, y_i) and (x_i, z_i) sentence pair respectively.

To compute the semantic similarity of two sentences in different languages, (Xu et al., 2019) relies on cosine similarities of sentence embedding vectors in a common vector space such as bilingual word embedding (Luong et al., 2015b). Our method first also embeds words in different languages into a common vector space as work in (Conneau et al., 2017), then calculate the sentence similarity based on *word alignment scores* and the *longest parallel phrase* of the candidate sentence pairs. In order to acquire word alignments of a sentence pair (x, y), we iterate sentence x from left to right and greedily align each word in x to the most similar word in y which was not already aligned. For measuring the similarity of words we use cosine similarity of word embeddings. Afterward, given a set of word alignments A, we can easily extract parallel phrases of (x, y) by using the phrase extraction algorithm in the Statistical Machine Translation System (Koehn et al., 2003). Finally, the semantic similarity score of the sentence pair (x, y) is computed by averaging word alignment scores and weighting it with the ratio of the length of the longest parallel phrase p and the length of the sentence x as follows:

$$\text{sim}(x, y) = \frac{|p|}{|x|} \times \frac{\sum_{a \subset A} \text{score}(a)}{|A|} \tag{10}$$

where $|p|$ and $|x|$ are the length of longest parallel phrase and sentence x respectively, $|A|$ is the number of word alignments, a is a word alignment candidate and $score(a)$ is word alignment score that is computed by using cosine similarity of two words in the alignment a.

5 Experiments

5.1 Dataset

In this work, we evaluate our approach on zero-resource Khmer-Vietnamese (km-vi) and Indonesian-Vietnamese (id-vi) language pairs with English is the pivot language. The parallel datasets for Khmer-English (km-en) and Indonesian-English (id-en) are from the Asian Language Treebank (ALT) Parallel Corpus (Riza et al., 2016) and for English-Vietnamese is from the UET dataset (Vu Huy et al., 2013) (see Table 1 for details). All testing datasets are from the ALT corpus with size of 1,018 sentences. In addition, we used monolingual data released in Wikipedia[1] for Vietnamese, English and Indonesia and data from WMT2020[2] for Khmer. After de-duplication and removing too short (<5 tokens) or too long (>100 tokens) sentences, we obtained approximately 11 million, 5 million, 2 million and 3 million unique sentences for English, Vietnamese, Khmer, and Indonesian respectively. Moreover, as mentioned in Section 4.1, before training models, we augmented the multilingual training data by using back-translation and self-training. In order to choose the right ratio between real and synthetic parallel data, we experimented on different real-to-synthetic ratios. We found that 1:4 real-to-synthetic ratio is the best ratio for both Khmer-English and Indonesian-English pairs as shown in Table 2. Finally, we acquired the

[1] https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/

[2] http://www.statmt.org/wmt20/parallel-corpus-filtering.html

Direction	Training	
	real	real+BT+ST
Khmer-English	18,088	162,792
English-Vietnamese	233,000	-
Indonesian-English	18,088	162,792

Table 1: Number of sentences used for training. $real$ column show the size of original data and $real$+BT+ST column illustrates the size of the augmented data.

real:syntheic ratio	km → en		id → en	
	BT	ST	BT	ST
1:0	14.19	13.7	21.57	20.52
1:1	15.32	15.72	22.26	21.18
1:2	16.87	17.58	24.06	22.10
1:3	17.25	18.21	24.37	21.99
1:4	**18.3**	**18.62**	**24.79**	**22.64**
1:5	18.1	17.93	24.02	21.60
1:6	17.54	17.01	23.70	21.42

Table 2: Experiment results on BLEU score to choose the right real:synthetic ratios for Khmer→English (km→en) and Indonesian→English (id→en) using back-translation (BT) and self-training (ST).

final augmented data by combining the original data with back-translated and self-trained data as shown in Table 1. Note that, to prevent imbalances between language pairs in the multilingual training data, we did not augment for the English-Vietnamese pair since the size of this pair is much larger other pairs.

5.2 Preprocessing

To learn a shared vocabulary for training multi-NMT, we used SentencePiece (Kudo and Richardson, 2018) with size 32,000 over the combined English, Vietnamese, Khmer, and Indonesian monolingual data. Besides, we added target language tags at both the beginning and end of the source sentences in the multilingual training data.

The multilingual word embedding model used in our filtering method was acquired by using the unsupervised method in MUSE library[3]. The word embeddings for English, Vietnamese, Khmer, and Indonesian are trained with fastText toolkit[4] on corresponding monolingual data.

All translation results shown in our work were computed in terms of BLEU score (Papineni et al., 2002) measured with *multi-bleu.perl* script[5]

5.3 Models

All models in our experiments are based on the encoder-decoder with attention architecture (Luong et al., 2015a). We used OpenNMT-py[6] to run all experiments with the configuration as follows. We used the Gradient Descent optimizer with a learning rate of 1.0 that decayed exponentially in the last 80% of the training duration, training batch is 64, maximum sentence length is 100, beam width is 10, label smoothing is 0.2, dropout is 0.3 and is applied on top of various process, all models variables are initialized uniformly in range (-0.1, 0.1).

In this paper, we evaluate our proposed method on two direct (zero-shot) translations, Khmer → Vietnamese (km → vi) and Indonesian → Vietnamese (id → vi). Notably, the setting of experiments for these 2 directions is the same, so in the following, we only describe experiments for evaluating the Khmer-Vietnamese language pair.

Firstly, We compare our models to three baselines as follows:

- **zero-shot NMT**: This model is trained on the Khmer ↔ English and English ↔ Vietnamese parallel data.
- **zero-resource NMT**: This model is trained on the synthetic data Khmer ↔ Vietnamese created by using the above zero-shot NMT model to translate English sentences in (English ↔ Vietnamese) to Khmer sentences.
- **pivot language**: use the above zero-shot NMT to translate Khmer sentences into English then from English to Vietnamese.

Our proposed models are designated as below:

- **Iterative multi-NMT**: This model is trained by iterating *training-generating-training* schema for several rounds. We use the above zero-shot NMT as an initial multi-NMT model for this training process.

[3] https://github.com/facebookresearch/MUSE

[4] https://fasttext.cc/

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

[6] https://github.com/OpenNMT/OpenNMT-py

- **Iterative multi-NMT + Xu's data filtering**: This model is trained by iterating *training-generating-filtering-training* schema for several rounds as shown in Section 4. We also use the above zero-shot NMT as an initial multi-NMT model and use the method of (Xu et al., 2019) in the data filtering step.
- **Iterative multi-NMT + our data filtering**: This model is trained by using the *training-generating-filtering-training* process and our proposed method for data filtering.

Note that, in the last two models, we use a similarity threshold of 0.4 achieved the best result (see Table 4 for details), to filter out poor synthetic sentence pairs.

5.4 Results and Analysis

Table 3 shows our results for the $km \rightarrow vi$ and $id \rightarrow vi$ zero-resource translation experiments. Experiments (1), (2), and (3) indicate the performance of the three baseline models. It can be seen that zero-shot NMT performed the worst result while the two other models illustrate promising results. The explanation for this results is that the amount of multilingual training data is not enough for enabling zero-shot translation on the multi-NMT system. Experiment (4) outperforms all three baseline models since it is benefit from both zero-shot and zero-resource NMT system. In addition, Experiments (5) and (6) show the effect of our *training-generating-filtering-training* process. By eliminating poor synthetic sentence pairs before re-training, the systems perform better results. Especially, the results on experiment (5) and (6) indicate that our proposed filtering method is more effective than the method of (Xu et al., 2019) for filtering noises in synthetic data.

	model	**km→vi**	**id→vi**
(1)	zero-shot NMT	3.43	6.75
(2)	zero-resource NMT	13.82	14.26
(3)	pivot language	12.59	12.99
(4)	Iterative multi-NMT	15.23	17.24
(5)	Iterative multi-NMT + Xu's data filtering	16.02	18.51
(6)	Iterative multi-NMT + our data filtering	**16.87**	**18.93**

Table 3: BLEU scores for our proposed models compared with strong baselines.

Threshold	**km → vi**	**id → vi**
0.0	15.23	17.24
0.1	15.81	17.75
0.2	16.02	17.96
0.3	16.25	18.29
0.4	**16.87 (+1.64)**	**18.93 (+1.69)**
0.5	16.62	18.58
0.6	16.37	18.34

Table 4: The effect of the quality of filtered synthethic data with different filtering thresholds in terms of BLEU sore.

Table 4 shows the effect of different filtering threshold on translation performance. All models are trained similar to the model *Iterative multi-NMT + our data filtering*, the only different is the filtering threshold to eliminate poor sentence pairs. Notably, a threshold of 0.0 means that all synthetic data is kept to re-train in the next iteration. The results illustrate that the threshold of 0.4 achieved the best result, outperforming the baseline (threshold is 0.0) by +1.64 and +1.69 BLEU for $km \rightarrow vi$ and $id \rightarrow vi$ directions respectively.

On the other hand, Table 5 shows that if we fine-tune our proposed model *Iterative multi-NMT + our data filtering* on a small amount of real parallel data, the model performs a significant improvement by +9.26 and +4.76 over the baselines (models are only trained on real parallel data). The real datasets for $km \rightarrow vi$ and $id \rightarrow vi$ are from the ALT corpus with size of 18,088 sentence pairs. This results prove that our proposed model work well on both zero-resource and low-resource language pairs.

model	**km → vi**	**id → vi**
direct	13.39	16.81
Iterative multi-NMT + our data filtering + incremental training	**22.65 (+9.26)**	**21.57 (+4.76)**

Table 5: Translation performance (BLEU) when fine-tuning our proposed model on a small amount of real parallel data.

6 Conclusion

In this paper, we have shown a *training-generating-filtering-training* cycle to build a model for translating zero-resource language pairs. In addition, we proposed a simple filtering method based on word alignments and the longest parallel phrase to filter out poor quality sentence pairs from the synthetic data. Experiment results show that our proposed methods outperformed some strong baselines and achieve a promising result under zero-resource conditions for the Khmer→Vietnamese and Indonesian→Vietnamese directions. Specially, our proposed model can easily improve their quality with a small amount of real parallel data.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The missing ingredient in zero-shot neural machine translation, 03.

Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *CoRR*, abs/1710.04087.

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. pages 99–107, 01.

Yaroslav Gani, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario Marchand, and Victor Lempitsky. 2015. Domain-adversarial training of neural networks. 05.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. pages 1258–1268, 01.

Thanh Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. 11.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, November.

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.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. 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, November. Association for Computational Linguistics.

Surafel Melaku Lakew, Quintino Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Improving zero-shot translation of low-resource languages. 12.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective Approaches to Attention-based Neural Machine Translation. *CoRR*, abs/1508.04025.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado, June. 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, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

H. Riza, M. Purwoadi, Gunarso, T. Uliniansyah, A. A. Ti, S. M. Aljunied, L. C. Mai, V. T. Thang, N. P. Thai, V. Chea, R. Sun, S. Sam, S. Seng, K. M. Soe, K. T. Nwet, M. Utiyama, and C. Ding. 2016. Introduction of the asian language treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 12.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. 08.

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, August. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Hien Vu Huy, Phuong-Thai Nguyen, Tung-Lam Nguyen, and M.L Nguyen. 2013. Bootstrapping Phrase-based Statistical Machine Translation via WSD Integration. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1042–1046, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. 09.

Guanghao Xu, Youngjoong Ko, and Jungyun Seo. 2019. Improving Neural Machine Translation by Filtering Synthetic Parallel Data. *Entropy*, 21(12):1213, Dec.

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, November. Association for Computational Linguistics.