A Framework for Low Resource Language Translation based on SMT and Highly Accurate Word Alignment

Jingyi Zhu[1] Yizhen Wei[1] Takuya Tamura[1] Takehito Utsuro[1] Masaaki Nagata[2]
[1]Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba
[2]NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract

In this paper, we present a framework using SMT and highly accurate word alignment for low resource language translation. We align words in a parallel sentence pair using SpanAlign, a highly accurate word aligner based on cross-language span prediction. Then, we build bilingual phrase tables based on the alignments. For the SMT input, we use both normal-order sentences and pre-ordered sentences. The pre-ordering process is through an order-transform neural network called Pointer Network. The results on the Asian Language Treebank datasets show that the proposed SMT based on high precision alignment outperforms an NMT based on the Transformer in a simulated low resource translation setting using 20,000 parallel sentence pairs.

1 Introduction

Despite Neural Machine Translation(NMT) having achieved a state of the art performance in recent years, it is known as data-driven [1]. To overcome the challenge that exists in small-scale translation or low resource translation tasks, serval researches focus on the approaches such as pre-training with large scale corpus and fine-tuning with small-scale corpus [2], or map the monolingual vector embeddings into a common cross-lingual embedding space [3] [4]. However, these effective methods need a lot of computation [2] or large-sized parallel corpus.

In this paper, we propose a framework based on SMT and highly accurate word alignment to explore the feasibility of low resource language translation. Specifically, the framework does not need a sequence-to-sequence NMT model but uses phrase-by-phrase translation instead. Since we focus on the limitation of the low resource language corpus, we use the Asian Language Treebank corpus, which contains 20,000 parallel sentences as the base corpus. We do the experiments between the directions of the Japanese-English pair and the Japanese-Chinese pair. We experimentally show that our proposed framework outperforms an NMT based on the Transformer.

2 Related Work

Although pre-ordering has often been used in SMT-related works, some research has recently applied pre-ordering to NMT. Kawara et al. [5] discussed the influence of word order on the NMT model, and concludes that it is important to keep the consistency between the input source word order and the output target word order, to improve the translation accuracy. Murthy et al. [6] proposed a transfer learning approach for NMT, which trains an NMT model on an assisting language-target language pair, improves the translation quality in extremely low-resource scenarios. Nevertheless, those methods both rely on the neural network translation model or separately pre-training a translation model by a large-scale corpus. In contrast, our proposed framework has no neural translation component and we focus on the translation task limited by a small-scale corpus.

3 The Framework based on SMT and Word Alignment

This section mainly demonstrates the whole process of the proposed framework, as shown in Figure 1.

First, We fine-tune multilingual BERT using the manually made word alignment data, then we use the word alignment model to align words in the training sentences. The word alignment data is used to train the Moses model, consisting of the phrase table and language model. At last, the original order test data or preordered test data is translated phrase-by-phrase. On the other hand, Figure 1(b) shows the procedure to create pre-ordered test data. The word alignment of the training corpus is also used to train the Pointer Network. Then the trained Pointer Network transforms the original order test data into pre-ordered test

data.

3.1 Word Alignment by SpanAlign

SpanAlign [7] is a multilingual BERT [8] based alignment method, which formalizes a word alignment problem as a collection of independent predictions from a token in the source sentence to a span in the target sentence. Because our method relies on high precision alignments to make bilingual phrase tables, and training data for Pointer Network, we apply SpanAlign to extract the alignments from the parallel corpus.

3.2 Pre-ordering by Pointer Network

The pre-ordering process transforms the orders of the tokens in a source sentence to those of the tokens in its target sentence before translation is performed. Figure 2 shows an example for transferring Japanese sentence.

The original Pointer Network is an LSTM [9] based neural network, which aims at solving graph theory problems such as the Traveling salesman problem and Convex Hull. An encoding RNN converts the input sequence to a code (blue) that is fed to the generating network (purple) [10]. At each step, the generating network produces a vector that modulates a content-based attention mechanism over inputs. The output of the attention mechanism is a softmax distribution with a dictionary size equal to the length of the input.

Inspired by this, we apply Pointer Network to word order rearrangement like Figure 3. Specifically, we replace the input of Pointer Network with a sequence of the token instead, and then add an embedding layer to represent words with vectors. At decoding time, the decoder predict next pointer probability $p(C_i|C_1, ..., C_{i-1}, P)$ rely on inputs and predicted outputs :

$$u^i_j = v^T tanh(W_1 e_j + W_2 d_i) \quad j \in (1, ..., n) \qquad (1)$$

$$p(C_i|C_1, ..., C_{i-1}, P) = softmax(u^i) \qquad (2)$$

where softmax normalizes the vector u^i (of length n) to be an output distribution of inputs. P is the input sentence, and C_i is the token of output sentence. u^i is the vector. Parameters v, W_1, W_2 are learnable parameters of the output model, and e_j, d_i represents for the encoder state and decoder state, respectively.

3.3 Bilingual Phrase Table based Phrase-by-Phrase Translation

Bilingual phrase tables are lists of terms (words or phrases) in one language associated with their translations in a second language. Therefore, Phrase-by-Phrase translation is a process that, for each token in the source sentence, retrieve and output the most appropriate target tokens in the built-up phrase table. In our proposed framework, we replace the GIZA++ [1)] which is contained in Moses with SpanAlign, to evaluate whether the improvement of alignment accuracy has an impact on the statistical machine translation.

4 Experiments

4.1 Dataset

We use the ALT (Asian Language Treebank)[2)] as our main experiment corpus. Here we use English-Japanese and Chinese-Japanese, about 20K sentence pairs for each language pair. Parallel data are divided into the training data (18K) and the test data (1K). We use Japanese-English and Chinese-Japanese corpus because the word-order divergence of the two languages is very large and manually made word alignment data is available. We use MeCab[3)] and Jieba[4)] to tokenize Japanese and Chinese sentences into tokens, respectively. The English side is tokenized by tokenizer.perl in the Mosesdecoder.

4.1.1 SpanAlign Settings and Fine-tuning

We use the ALT Japanese-English dev data of about 1,000 sentences of word alignment data to fine-tune SpanAlign for Japanese-English. For Chinese-Japanese, we use about 3,000 sentences of word alignment data created by NTT to fine-tune SpanAlign. We follow the parameter as default, while the training batch size is set to 8 and the training epoch is 10. The average extraction threshold in bidirectional sides is 0.4.

4.1.2 Pointer Network Settings and Training

Training data for the Pointer network are the training data of original order sentences and pre-ordered sentences made by the alignments generated by SpanAlign. We use a

1) https://github.com/moses-smt/giza-pp
2) https://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/
3) https://github.com/neologd/mecab-ipadic-neologd
4) https://github.com/fxsjy/jieba

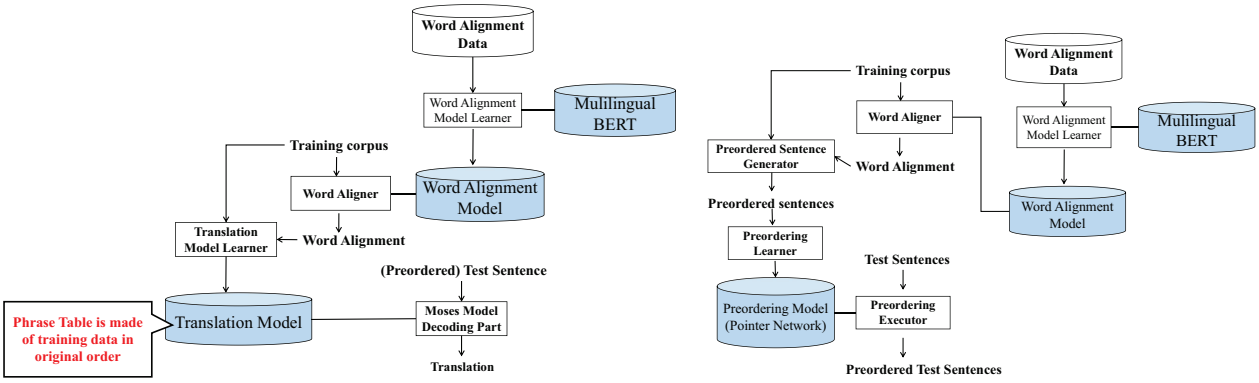

Figure 1 Our proposed Framework: (a) is the flow chart, which accepts the normal order sentence or preordered sentence as translation input, (b) is using Pointer Network to do the preordering.

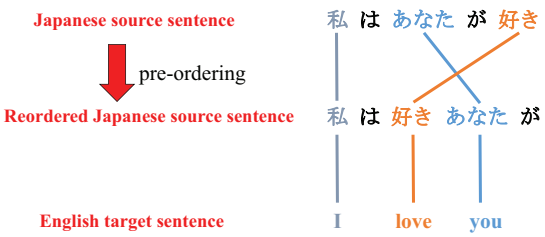

Figure 2 Transform the word order of the source Japanese language to the target English language

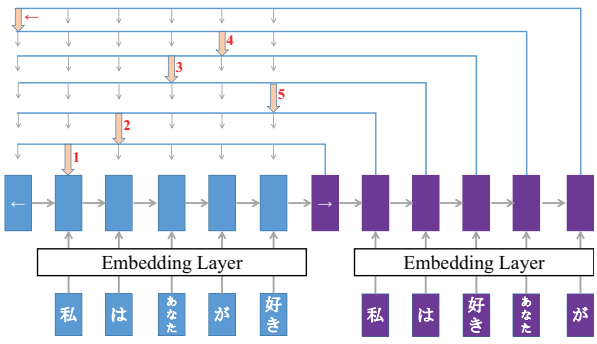

Figure 3 Architecture of Pointer Network (the modified Pointer Network accepts the original order sequence as input, and outputs the pre-ordered sequence).

2-layer bidirectional LSTM, with a hidden state of 512 and an embedding state of 128. And we set the training batch size to 16, the learning rate to 3e-4, the training epoch is 10, max sequence length to 120. After training, the weighted Pointer Network is used to do the pre-ordering operation for test data sentences. We exploit RIBES [11], an efficient measure for automatically evaluating machine translation qualities based on the order of words, to evaluate the performance of the Pointer Network.

4.1.3 Phrase-by-Phrase Translation

We use Moses[5] to make the phrase table, and the maximum length of each phrase is set to 3. The difference between our framework and previous pre-ordering of SMT is that we use the original order data pairs to make the phrase table, and we only apply preordering to the test data. We use a trigram LM (Language Model), which is learned by target side sentences contained in the training-part corpus, to ensure the fluency of the output language. For the Japanese to English direction and Chinese to Japanese direction, we compare the translation results between alignments from SpanAlign and Awesome-align [12], which is also based on a pre-trained multilingual language model but does not require manually made word alignment data.

4.2 Results

4.2.1 Pointer Network Performance

Because there is no ALT Chinese-Japanese manual alignment data exist for evaluation, we only use Japanese

5) https://www.statmt.org/moses/

Table 1 BLEU score between baseline and proposed approach.

Model	Direaction	Alignment Approach	SMT Input Order	PT Size	BLEU
Transformer	Ja → En	-	-	-	8.12
Phrase-by-phrase + LM	Ja → En	SpanAlign	Original	495853	**9.23**
Phrase-by-phrase + LM	Ja → En	SpanAlign	Pre-order	495853	8.74
Phrase-by-phrase + LM	Ja → En	Awesome-align	Original	1038614	8.58
Phrase-by-phrase + LM	Ja → En	Awesome-align	Pre-order	1038614	7.99
Transformer	Zh → Ja	-	-	-	6.14
Phrase-by-phrase + LM	Zh → Ja	SpanAlign	Original	418647	8.24
Phrase-by-phrase + LM	Zh → Ja	SpanAlign	Pre-order	418647	**10.11**
Phrase-by-phrase + LM	Zh → Ja	Awesome-align	Original	959425	7.95
Phrase-by-phrase + LM	Zh → Ja	Awesome-align	Pre-order	959425	8.55
Transformer	En → Ja	-	-	-	5.91
Phrase-by-phrase + LM	En → Ja	SpanAlign	Original	495853	9.83
Phrase-by-phrase + LM	En → Ja	SpanAlign	Pre-order	495853	**11.61**
Transformer	Ja → Zh	-	-	-	4.08
Phrase-by-phrase + LM	Ja → Zh	SpanAlign	Original	418647	**8.36**
Phrase-by-phrase + LM	Ja → Zh	SpanAlign	Pre-order	418647	7.17

and English data to verify the performance of the Pointer Network. Table 2 shows the F1 score between SpanAlign and Awesome-align, demonstrating the high alignment accuracy. Table 3 shows the result of the score of the pre-ordered test data for transferring Japanese order into English order verified by RIBES. Here, we see ALT Japenese manual alignment data as the reference. From the results, Pointer Network trained with tokens extracted from SpanAlign is nearly the same as that of manual alignment. Thus, it can be considered that Pointer Network successfully learned certain language order features which are effective for the pre-ordering task.

Table 2 F1 score of SpanAlign and Awesome-align

	P	R	F1
Awesome-align	0.71	0.46	0.56
SpanAlign	**0.79**	**0.86**	**0.83**

Table 3 RIBES result of Pointer Network trained by each approach, of transferring Japanese order into English order

	RIBES
Manual alignment	0.761
Awesome-align	0.623
SpanAlign	**0.751**

4.2.2 Translation accuracy

As a criterion to verify the translation accuracy, we use the BLEU [13] score. And we select Transformer [1] as our baseline. Table 1 shows the accuracy of our proposed method and the accuracy of the baseline. The results show that the proposed method exceeds the NMT model in every setting. Note that we did not use mert to fine-tune any weight of the translation model and language model. However, for the direction of Japanese, better results were obtained using the original order language as input.

We also tried making the phrase table after reordering the training data, however, the BLEU score is lower than that made by original order data.

5 Conclusion and Future Work

In this paper, we proposed a framework for low resource translation without using the sequence-to-sequence neural translation model. We use the normal-order tokens and the pre-ordered tokens as input and translated phrase-by-phrase. The results show that both methods exceed the baseline of NMT for high precision alignment. And as the accuracy of alignment extraction increases, the accuracy of translation also increases. In future work, we will continue our experiments with other language pairs, like southeast Asian languages.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, U. Kaiser, and I. Polosukhin. Attention is all you need. In **NIPS**, 2017.

[2] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. Multilingual denoising pre-training for neural machine translation. **Transactions of the ACL**, pp. 726–742, 2020.

[3] Z. Lin, X. Pan, M. Wang, X. Qiu, J. Feng, H. Zhou, and L. Li. Pre-training multilingual neural machine translation by leveraging alignment information. In **proc. EMNLP**, pp. 2649–2663, 2020.

[4] S. Sen, K. Gupta, A. Ekbal, and P. Bhattacharyya. Mul-

tilingual unsupervised NMT using shared encoder and language-specific decoders. In **Proc. 57th ACL**, pp. 3083–3089, 2019.

[5] Y. Kawara, C. Chu, and Y. Arase. Recursive neural network-based preordering for statistical machine translation and its analysis. **Journal of Natural Language Processing**, Vol. 26, No. 1, pp. 155–178, 2019.

[6] R. Murthy, A. Kunchukuttan, and P. Bhattacharyya. Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. In **Proc. NAACL**, pp. 3868–3873, 2019.

[7] M. Nagata, K. Chousa, and M. Nishino. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In **Proc. EMNLP**, pp. 555–565, 2020.

[8] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proc. NAACL**, pp. 4171–4186, 2019.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. **Neural Computation**, Vol. 9, No. 8, p. 1735–1780, 1997.

[10] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In **NIPS**, 2015.

[11] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada. Automatic evaluation of translation quality for distant language pairs. In **Proc. EMNLP**, pp. 944–952, 2010.

[12] Z. Dou and G. Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In **proc. 16th EACL**, pp. 2112–2128, 2021.

[13] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proc. 40th ACL**, pp. 311–318, 2002.