Leveraging Highly Accurate Word Alignment for Low Resource Translation by Pretrained Multilingual Model

Jingyi Zhu s2120801_@_u.tsukuba.ac.jp
Minato Kondo s2320743_@_u.tsukuba.ac.jp
Takuya Tamura s2120744_@_u.tsukuba.ac.jp
Takehito Utsuro utsuro_@_iit.tsukuba.ac.jp
Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba, Japan

Masaaki Nagata masaaki.nagata_@_ntt.com
NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract

Recently, there has been a growing interest in pretraining models in the field of natural language processing. As opposed to training models from scratch, pretrained models have been shown to produce superior results in low-resource translation tasks. In this paper, we introduced the use of pretrained seq2seq models for preordering and translation tasks. We utilized manual word alignment data and mBERT-based generated word alignment data for training preordering and compared the effectiveness of various types of mT5 and mBART models for preordering. For the translation task, we chose mBART as our baseline model and evaluated several input manners. Our approach was evaluated on the Asian Language Treebank dataset, consisting of 20,000 parallel data in Japanese, English and Hindi, where Japanese is either on the source or target side. We also used in-house 3,000 parallel data in Chinese and Japanese. The results indicated that mT5-large trained with manual word alignment achieved a preordering performance exceeding 0.9 RIBES score on Ja-En and Ja-Zh pairs. Moreover, our proposed approach significantly outperformed the baseline model in most translation directions of Ja-En, Ja-Zh, and Ja-Hi pairs in at least one of BLEU/COMET scores.

1 Introduction

In recent years, there has been a growing body of research on sequence-to-sequence (seq2seq) models that are based on pretraining (Xue et al., 2021; Liu et al., 2020; Lin et al., 2020). Since the introduction of the Transformer architecture (Vaswani et al., 2017), the quality of machine translation has greatly improved. However, when it comes to low-resource translation tasks, the performance of this type of parameter randomization model often suffers due to the limited size of available datasets (Sennrich and Zhang, 2019; Lee et al., 2022; Zhu et al., 2022).

To address this challenge, many researchers have proposed using unsupervised methods, such as mapping monolingual vector embeddings to a common cross-lingual embedding space (Lin et al., 2020; Sen et al., 2019), or leveraging large-scale pretraining models that have been successfully applied to various NLP tasks (Devlin et al., 2019; Brown et al., 2020).

Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track, pages 336–347
September 4–8, 2023, Macau SAR, China.

In this paper, we propose applying a pretrained seq2seq model for preordering and translation tasks. Specifically, we investigate different sizes of mT5s (Xue et al., 2021) and mBART (Liu et al., 2020), in order to evaluate their performance on preordering when using manual word alignment data. For the translation process, we choose mBART as our baseline model, and we evaluate the translation results using both the original sequence and the generated preordering sequence as input. Our approach was evaluated on the Asian Language Treebank dataset (Riza et al., 2016), consisting of 20,000 parallel data in Japanese, English and Hindi, where Japanese is either on the source or target side. To compare the effects on different datasets, we also used the in-house data which comprised 3,000 parallel data in Chinese and Japanese. The results indicated that mT5-large trained with manual word alignment achieved a preordering performance exceeding 0.9 when evaluated using the RIBES score on Ja-En and Ja-Zh pairs. Moreover, our proposed approach significantly outperformed the baseline model in most of the translation directions of language pairs of Ja-En, Ja-Zh, and Ja-Hi in terms of at least one of the BLEU(Papineni et al., 2002) and COMET (wmt20-comet-da) (Rei et al., 2020) scores.

2 Related Work

In recent years, researchers have conducted more and more studies on seq2seq models based on pretraining. While learning the rules of sequence generation remains the most crucial feature of these models, some studies have explored the application of preordering to training, resulting in improved results. Kawara et al. (2018) discussed the importance of maintaining consistency between input source word order and output target word order for improved translation accuracy in neural machine translation (NMT) models. Murthy et al. (2019) proposed a transfer learning approach for NMT that trains the model on an assisting source-target language pair and improves translation quality in extremely low-resource scenarios. However, both methods rely on separately pretraining a translation model using a large-scale parallel corpus and handle preordering based on the syntax tree. In contrast, Zhu et al. (2022) proposed a framework for low-resource translation that focuses on preordering and highly accurate word alignment using an SMT model. Their solution outperformed the Transformer model, but they did not explore the use of large-scale pretrained seq2seq models.

Our work focused on low-resource translation tasks and utilizes large-scale pretrained multilingual models for fine-tuning the preordering and translation procedures.

3 Seq2seq Models

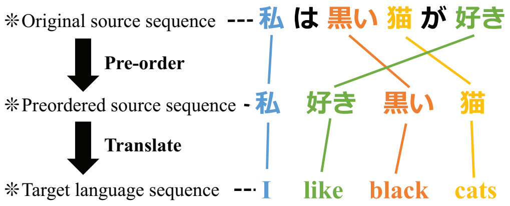
In general, seq2seq models take a sequence of tokens as input from the source sequence $S = s_1, s_2, \ldots, s_k$ and produce a sequence of tokens as output for the target sequence $T = t_1, t_2, \ldots, t_m$, where $s_i(i = 1, \ldots, k)$ and $t_j(j = 1, \ldots, m)$ represent the tokens in the source and target sequences, respectively.

In terms of structure, seq2seq models consist of an encoder and a decoder (Hochreiter and Schmidhuber, 1997). The encoder converts the input sequence into a high-dimensional vector representation, while the decoder maps the high-dimensional vectors to the output dictionary based on the encoder's output. This framework has been applied to various tasks, including machine summarization (Shi et al., 2021), question-answering systems (Yin et al., 2016), and machine translation (Sutskever et al., 2014). Since seq2seq models can learn the rules governing the input and output sequences, we aim to use them for preordering and translation.

4 Seq2seq Models for Preordering

4.1 Preordering Process

While preordering is commonly utilized in statistical-based translation systems, it is also possible to implement preordering in seq2seq systems. The preordering procedure entails arranging the tokens in a source sequence to those of the tokens in its target sequence before translation is performed. An example of transferring a Japanese sentence is shown in Figure 1.



※Original source sequence
私 は 黒い 猫 が 好き
Pre-order
※Preordered source sequence
私 好き 黒い 猫
Translate
※Target language sequence
I like black cats

Figure 1: Transform the word order of the source Japanese language to the target English language before translation.

Regarding the preordering procedure, we use mT5 (Xue et al., 2021) and mBART (Liu et al., 2020), which are kinds of state-of-the-art seq2seq models. Both models have encoder-decoder structures based on self-attention, with a minor variation in their pretraining tasks.

4.2 Reordered Training Data

As our preordering method is entirely based on the seq2seq model, it is necessary to construct the required training data, which is produced by manual word alignment data.

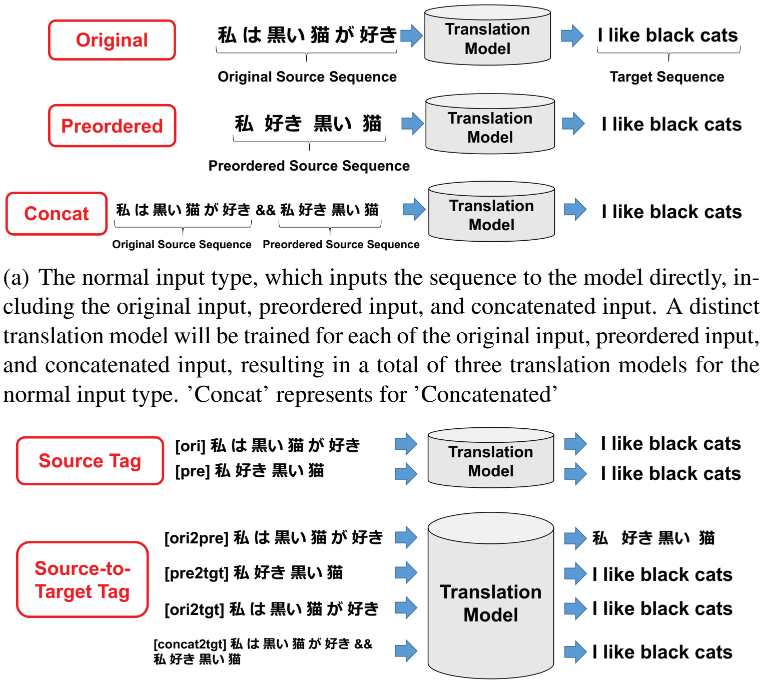
Formally, word alignment can be defined as: given a sentence $X = \{x_1, x_2, ..., x_m\}$ in the source language and its corresponding parallel sentence $Y = \{y_1, y_2, ..., y_n\}$ in the target language, the word alignment are set of pairs of source and target words using the following equation:

$$Alignment = (< x_i, y_j >: x_i \in X, y_j \in Y) \qquad (1)$$

The aligned pair of words x_i and y_j are semantically similar within the context of the sentence.

Having those word alignments, for the model input, we use the original source sequence. On the output side, we simply ignore the NULL-aligned tokens, which were not aligned with any tokens on the target side. For instance, the Japanese sentence ” 私 (I) は 黒い (black) 猫 (cat) が 好き (like) ” can be easily preordered into the English order of ” 私 (I) 好き (like) 黒い (black) 猫 (cat)” with the alignments of (私-*I*), (黒い-*black*), (猫-*cat*), and (好き-*like*) based on the word alignment. Therefore, we ignore ” は” and ” が” in the preordered sequence because they were not aligned to any tokens. After removing ” は” and ” が” from the output side of the preordered sequence, the training pair becomes ” 私 (I) は 黒い (black) 猫 (cat) が 好き (like)” and ” 私 (I) 好き (like) 黒い (black) 猫 (cat)”. We use such training pairs to train order transformation seq2seq neural networks.

We utilized two types of word alignment data to generate our training data. The first type is based on manual word alignment data, while the second type is derived from the word alignment data generated by mBERT (Devlin et al., 2019). To automatically extract word alignment from parallel corpus data, we employed the AWESoME-align (Dou and Neubig, 2021), which is capable of unsupervised fine-tuning by adjusting the embedding distribution of the output from a multilingual BERT in order to achieve accurate word alignments. One significant advantage of this approach is that it eliminates the need for manual word alignment data.



(a) The normal input type, which inputs the sequence to the model directly, including the original input, preordered input, and concatenated input. A distinct translation model will be trained for each of the original input, preordered input, and concatenated input, resulting in a total of three translation models for the normal input type. 'Concat' represents for 'Concatenated'

(b) The tagged input type, which places the unique tag before each input sequence, including source tag input and source-to-target tag input. A total of two translation models will be trained for the tagged input type, where one model is trained for each source tag input and source-to-target tag input.

Figure 2: Two input types of **(a) Normal Input** and **(b) Tagged Input**.

5 Seq2seq Model for Translation

5.1 Training Pattern

We utilize mBART as the primary translation model for the translation process. In order to compare the results of several input variations, we experimented with various "training patterns", consisting of the normal input type and the tagged input type. Normal input type refers to sequences directly fed into the model, including the original input, preordered input, and concatenated input, as shown in Figure 2 (a). On the other hand, tagged input type includes a sequence type tag at the beginning of each sequence, which includes the source tag input and the source-to-target tag input as shown in Figure 2 (b).

For normal input type, we trained translation models using the original input, preordered input, and concatenated input separately. In other words, we trained three models and tested the translation accuracy of each pattern of input. Original input uses the original source language sequence as input and outputs the target language sequence as shown in "original" of Figure 2 (a). We see this pattern of the input as the seq2seq translation baseline.

- **Original input**: Original source sequence ⇒ Target language sequence

In order to verify whether the utilization of preordering in isolation can result in an enhancement of translation accuracy, we use the preordered source language sequence as input and output the corresponding target language sequence as shown in "preorder" of Figure 2 (a).

- **Preordered input**: Preordered source sequence ⇒ Target language sequence

In addition to this, we also attempted to use a concatenation approach by combining the sequences of the original and preorder together and splitting them using learnable symbols as shown in "concat" of Figure 2 (a). This kind of input is intended to leverage the information from both the original sequence and the preordered sequence.

- **Concatenated input**: Original source sequence && Preordered source sequence ⇒ Target language sequence

For tagged input, we aim to verify whether the translation accuracy improves by increasing the amount of training data. To achieve this, we differentiated the input types by mixing original, preordered, and concatenated sequences, while each sequence is prefixed with a corresponding tag to facilitate this process. For each of the source tag input and the source-to-target tag input, a separate translation model is trained respectively. In other words, for the tagged input type, a total of two models are trained. Source tag input uses both original and preordered source language sequences as input but carries the sequence type tag at the head of the sequence (for example, using [ori] and [pre] to represent the original sequence and preordered sequence) as shown in "source tag" of Figure 2 (b). It stands to reason that the actual amount of training data is twice the baseline due to the mixture of inputs from the original and preordered sequences.

- **Source tag input**: [ori] Original source sequence ⇒ Target language sequence

  [pre] Preordered source sequence ⇒ Target language sequence

In addition to the previously mentioned training mode, we also experimented with a source-to-target tag input to maximize the amount of training data using our method as shown in "source-to-target tag" of Figure 2 (b). This training pattern combines four inputs: from original source sequence to preordered source sequence, from original source sequence to target language sequence, from preordered source sequence to target language sequence, and from concatenated sequence to target language sequence. To enable the model to distinguish between the types of input and output corresponding sequences, we added tags [ori2pre], [pre2tgt], [ori2tgt], and [concat2tgt] to each kind of sequences, respectively. The reason we tried this input method is that, unlike the source tag input, which only outputs from the source language sequence to the target language sequence, the process of learning preordering is added during the translation model training, allowing the model to more appropriately learn the rules for generation from the source language sequence to the target language sequence. Note that the training data of source-to-target tag input is fourth the baseline because we mixed four kinds of inputs and outputs.

- **Source-to-target tag input**: [ori2pre] Original source sequence ⇒ Preordered source sequence

  [ori2tgt] Original source sequence ⇒ Target language sequence

  [pre2tgt] Preordered source sequence ⇒ Target language sequence

  [concat2tgt] Original source sequence && Preordered source sequence ⇒ Target language sequence

5.2 Test Pattern

In the generation stage, to each of the trained translation models described in the previous section, we input test data comprised of corresponding input pattern, which is referred to as "test pattern"[1].

[1] The correspondence between the training and test patterns are shown in the columns of "Training Pattern" and "Test Pattern" in Table 5.

Since each training pattern of the normal input type has its own translation model trained on its corresponding input data, we directly input the corresponding pattern of test data into the model to obtain the translation results. For example, we input the test data of the test pattern of the original input into the model, which is trained with the training pattern of the original input, to obtain the translation results.

During the training of the translation models of the tagged input type, it can be considered that we trained multiple models with different test patterns, inputs, which allows us to translate inputs of multiple test patterns simultaneously during testing.

For the source tag input, we input both the original and preordered sequences with tags during training, which enables us to evaluate the translation accuracy of the original or preordered sequences separately when conducting translation evaluation on the test set. For example, we add the [ori] tag before the original test data sequence, or the [pre] tag before the preordered test data sequence, and input either of them into the model trained with the training pattern of the source tag input to obtain the respective translation results.

For the source-to-target tag input, we simultaneously input the original, preordered, and concatenated sequences with tags during training. Therefore, when evaluating the translation accuracy of the test data, we can evaluate the translation accuracy of the original, preordered, or concatenated sequences separately. For example, we add the [ori2tgt] tag before the original test data sequence, the [pre2tgt] tag before the preordered test data sequence, or the [concat2tgt] tag before concatenated test data sequence and input either of them into the model trained with the training pattern of the source-to-target tag input to obtain the respective translation results. Although we added the process of generating the preordered sequence from the original sequence in the training process of source-to-target tag input, we did not add the accuracy of this process in our paper as we focused solely on the translation results[2]. The preordered sequence (which is evaluated through preordered input or concatenated input) used in testing source-to-target tag input is generated by mT5 rather than mBART.

6 Experiments

6.1 Dataset

In our seq2seq experiments, we utilized ALT[3] Japanese-XX (English and Hindi) and in-house Chinese-Japanese parallel datasets as our primary datasets. It is worth noting that in ALT, manual word alignment data is not available for language pairs other than Ja-En, so we only conduct manual word alignment on this pairs. The dataset was partitioned into training, validation, and test sets. Each subset of the ALT dataset contains 18K, 1K, and 1K parallel sequence pairs respectively, while the in-house dataset includes 2K, 0.5K, and 0.5K parallel sequence pairs. The amount of training data for each training pattern is presented in Table 1.

6.2 Preordering Setting

We created training data for seq2seq preordering by manual word alignment as described in Section 4.2. We compare preordering results using RIBES (Isozaki et al., 2010) between mT5-small, mT5-base, mT5-large and mBART-large (*mbart-large-50*)[4][5]. The preordered sequence

[2] We also evaluated the performance of preordering obtained through source-to-target tag input. However, the precision obtained was not as high as that obtained through mT5-large.

[3] `https://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/`

[4] All pretrained seq2seq models are downloaded from the public Huggingface library.

[5] Each model was trained for 40,000 steps with a training batch size of 16 and a learning rate of 3e-5. Additionally, we trained another mT5-large with a batch size of 32 because it achieved the best preordering result with a batch size of 16. We also attempted to train mT5-large with a batch size of 64, but the preordering result was lower than when training with a batch size of 32.

Input Type	Training Pattern	Training Data	
		Ja-XX	Ja-Zh
Normal	Original	18K	2K
	Preorder	18K	2K
	Concatenated	18K	2K
Tagged	Source	36K	4K
	Source-to-target	72K	8K

Table 1: The number of training data for each training pattern. 'XX' represents for English and Hindi.

Language Pairs	Precision	Recall	F1
Ja-En	0.79	0.60	0.68
Ja-Zh	0.84	0.68	0.75

Table 2: Precision, Recall, and F1 scores of AWESoME-align compared with manual word alignment in the language pairs of Ja-En and Ja-Zh.

was generated using the model with the maximum BLEU score against the validation set. The preordering process was executed on the NVIDIA RTX A6000 with CUDA 11.3.

For AWESoME-align, which automatically extracts word alignments, we only use the original parallel corpus to fine-tune due to its unsupervised nature. Furthermore, the parameters are not shared between different language pairs, meaning we fine-tune each language pair with a different instance of AWESoME-align. Regarding hyperparameters, we fine-tune each language pair for 10 epochs with a batch size of 16 and a learning rate of 3e-5. The accuracy of word alignments extracted by AWESoME-align has been presented in Table 2. In this table, manual word alignment is considered as the reference. However, since manual word alignment data is only available for Ja-En and Ja-Zh, we have reported the results only for those language pairs.

6.3 Translation Setting

We trained the mBART translation models using Fairseq[6]. Each model was trained for 40,000 steps with a maximum input length of 1,024 and a learning rate of 3e-5, which were the same as those used for the preordering process. We selected the model with the minimum label-smoothed cross-entropy loss during the generation stage on the validation set to generate the target translation. We used the preordered sequences generated by mT5-large, which was trained with a batch size of 32, as inputs for the mBART models. The translation process was executed on the NVIDIA RTX TITAN with CUDA 10.3.

7 Results

7.1 Preordering Performance

The RIBES columns in Table 3 display the comparison of RIBES scores for different seq2seq models to generate the preordered sequence. The results demonstrate that the RIBES score for mT5-large models trained with manual word alignment exceeds 0.9, regardless of the batch size used during training (i.e., 16 or 32). Table 4 reports the comparison of RIBES scores for transferring the original source sequence to the preordered source sequence using mT5-large when trained with manual word alignment or generated word alignment. Under our experimental conditions, the unigram precision is not equal to a hundred as the generated preordered sequence tends to include more tokens than the reference preordered sequence. It is obvious

[6] `https://github.com/facebookresearch/fairseq`

Preordering	# Parameters	Training	BLEU		RIBES	
Model		Batch Size	Ja⇒En	En⇒Ja	Ja⇒En	En⇒Ja
Oracle	-	-	34.82	34.27	-	-
mT5-small	300M	16	21.44	26.06	0.876	0.872
mT5-base	580M	16	24.38	27.68	0.895	0.889
mT5-large	1200M	16	24.83	28.48	0.901	0.905
		32	25.22	28.34	0.904	0.909
mBART-large	610M	16	23.28	27.28	0.883	0.894

Table 3: BLEU scores of using mBART as the translation model for translating Ja-En pairs when applying the preordered training/test pattern in normal input type among different preordering models, and RIBES results of seq2seq model trained by manual word alignments of transferring Japanese order into English order and opposite.

	Ja⇒En		En⇒Ja		Ja⇒Zh		Zh⇒Ja		Ja⇒Hi	Hi⇒Ja
Alignment used	M	A	M	A	M	A	M	A	A	A
RIBES	0.904	0.896	0.909	0.904	0.927	0.883	0.919	0.894	0.883	0.877
Unigram Precision	0.91	0.88	0.92	0.91	0.89	0.75	0.83	0.77	0.86	0.85
Normalized Kendall's Tau	0.93	0.93	0.94	0.93	0.96	0.96	0.97	0.96	0.92	0.92
Brevity Penalty	0.96	0.94	0.95	0.97	0.93	0.96	0.95	0.98	0.96	0.98

Table 4: RIBES scores when transferring the original source sequence to preordered source sequence using mT5-large. 'Alignment used' means manual word alignment or mBERT-based generated word alignment is used for training preordering. 'M' is short for 'Manual', while 'A' represents 'AWESoME'.

from the table that mT5-large models trained with manual word alignment outperformed those trained with generated word alignment.

7.2 Translation Performance

Table 4(a) illustrates the BLEU (Papineni et al., 2002) and COMET (wmt20-comet-da) (Rei et al., 2020) for each translation direction with different word alignments. Our proposed approach when trained with manual word alignment significantly outperformed the baseline model in most of the translation directions of language pairs of Ja-En, Ja-Zh, and Ja-Hi in terms of at least one of the BLEU and COMET (wmt20-comet-da) scores. Moreover, even with mBERT-based generated word alignment, our proposed approach significantly outperformed the baseline model in the translation directions of Ja to En, Zh to Ja, Ja to Hi, and Hi to Ja in terms of the COMET score. Our findings suggest that when utilizing manual word alignment as preordering training data, concatenated inputs exhibit the highest BLEU or COMET scores compared to other input patterns. However, when using the AWESoME word alignment, the original input mostly yields the best BLEU results, while concatenated inputs mostly generate the best COMET results. For the better results illustrated in concatenated inputs when using manual word alignment, we speculate that this could be due to the models learning the relative positions between the source and target languages by combining the original and more highly accurate preordered sequences as the concatenated input. When utilizing AWESoME word alignment, it is predictable that the preordered sequence contains more noisy positional information than manual word alignments. During the generation process, those erroneous positional information inevitably impact the output quality. However, due to the implementation of multiple input manners, the model could still achieve a higher precise output based on the original input. To conduct comparison experiments, we employed the **oracle** approach as shown in Table 4(b), which involves preordering the source test set according to the target test set using manual

(a) Preordering by mT5

Metrics					BLEU						COMET					
Alignment	Input	Training	Test	Preorder	Ja-En		Ja-Zh		Ja-Hi		Ja-En		Ja-Zh		Ja-Hi	
	Type	Pattern	Pattern	Model	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐
-	Normal	Original (Baseline)	Original (Baseline)	-	25.7	29.3	14.1	17.9	30.0	19.3	47.9	54.4	67.9	54.4	16.2	29.1
Manual	Normal	Preorder	Preorder	mT5	25.2	28.3	14.2	18.5†	-	-	47.9	51.0	63.2	53.2	-	-
		Concat	Concat	mT5	25.6	29.6	**14.7**†	19.2†	-	-	49.2	53.4	65.1	58.5	-	-
	Tagged	Source	Original	-	25.8	29.2	14.2	18.0	-	-	48.7	52.2	**66.2**	55.6	-	-
			Preorder	mT5	25.5	28.5	14.0	17.8	-	-	48.5	50.1	63.0	55.3	-	-
		S-T	Original	-	25.8	28.9	14.0	18.8†	-	-	46.8	52.7	64.3	58.4	-	-
			Preorder	mT5	25.1	28.2	13.7	18.3	-	-	46.4	51.0	61.5	57.3†	-	-
			Concat	mT5	**25.9**	**29.7**	14.4	**19.4**†	-	-	**49.8**†	**54.2**	63.5	**60.2**†	-	-
AWESoME	Normal	Preorder	Preorder	mT5	23.9	28.3	13.7	18.2	29.4	18.6	46.7	50.6	61.6	53.6	18.4	29.0
		Concat	Concat	mT5	25.7	**29.8**	14.0	**19.4**†	30.0	19.4	49.9†	53.2	**64.7**	53.6	18.1	**31.6**†
	Tagged	Source	Original	-	**26.1**	29.4	**14.3**	19.0†	**30.1**	19.6	**50.0**†	53.0	64.4	56.7	17.7	30.3
			Preorder	mT5	24.6	28.8	13.5	19.0†	**30.1**	19.0	46.3	50.1	61.7	56.7	**19.1**†	30.0
		S-T	Original	-	26.0	29.2	12.8	18.5†	28.8	**19.7**	46.9	52.7	63.4	57.0	13.4	30.2
			Preorder	mT5	24.1	28.2	12.0	18.6†	29.3	19.2	42.1	50.1	59.7	55.4	15.9	29.2
			Concat	mT5	25.8	29.6	12.9	18.9†	29.5	19.6	47.0	**54.3**	63.6	**57.8**†	15.6	31.3†

(b) Preordering by Oracle

Metrics					BLEU						COMET					
Alignment	Input	Training	Test	Preorder	Ja-En		Ja-Zh		Ja-Hi		Ja-En		Ja-Zh		Ja-Hi	
	Type	Pattern	Pattern	Model	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐	⇒	⇐
Manual	Normal	Preorder	Preorder	Oracle	34.8	34.3	17.2	22.7	-	-	54.5	56.1	67.3	62.0	-	-
		Concat	Concat	Oracle	35.5	35.7	17.8	22.9	-	-	56.3	59.1	69.8	65.1	-	-
	Tagged	Source	Preorder	Oracle	33.6	33.6	16.1	20.9	-	-	53.6	56.2	64.8	58.4	-	-
		S-T	Preorder	Oracle	33.7	33.5	15.9	20.8	-	-	52.6	55.8	63.8	61.3	-	-
			Concat	Oracle	35.0	35.0	16.1	22.0	-	-	55.8	58.9	67.4	63.4	-	-
AWESoME	Normal	Preorder	Preorder	Oracle	36.8	34.3	18.4	22.6	36.0	24.4	54.6	54.2	65.6	55.9	24.7	30.3
		Concat	Concat	Oracle	40.3	36.2	19.0	23.5	36.8	26.0	58.3	58.8	69.7	63.5	26.8	37.5
	Tagged	Source	Preorder	Oracle	36.6	33.9	17.2	21.9	35.4	23.7	52.2	54.8	64.3	60.3	23.9	30.0
		S-T	Preorder	Oracle	36.2	33.2	16.1	21.3	34.7	24.2	48.6	52.8	62.1	58.1	20.9	30.4
			Concat	Oracle	39.1	35.4	16.7	21.9	35.4	24.9	55.4	58.9	67.5	63.1	24.0	36.5

Table 5: BLEU and COMET scores between the different training/test patterns. The results are translated by mBART. ’mT5’ represents ’mT5-large’. ’Oracle’ represents preordering the source test set according to the target test set using manual or AWESoME word alignment data, instead of generating the preordered sequences using the seq2seq model. Results in bold indicate the best BLEU or COMET results in a specific translation direction using different word alignments. ’Concat’ represents ’concatenated’ and ‘S-T’ represents ‘Source-to-target’. † for a significant difference ($p < 0.05$) from the baseline.

or AWESoME word alignment data, instead of generating the preordered sequences using the seq2seq model. Although this result is not practical, it still demonstrates the potential of applying our preordering method to the seq2seq model.

Table 3 displays the BLEU scores of different models when using the preordered training/test pattern in normal input type for translating Ja-En pairs[7]. The translation quality coin-

[7] The number of model parameters are from `https://github.com/google-research/`

Japanese original sequence	彼 は 金曜日 の 夜 に サウス メルボルン の 停車場 から トラム を 盗ん だ こと でも 訴え られ て いる 。
English target reference sequence	He is also accused of stealing a tram on Friday night , from South Melbourne depot .
Oracle (manual) preordered sequence	彼 て いる でも 訴え られ でも 盗ん だ こと トラム に 金曜日 夜 から サウス メルボルン 停車場 。
Oracle (AWESoME) preordered sequence	彼 いる でも 訴え でも 盗ん を トラム に 金曜日 夜 から サウス メルボルン 停車場。
Generated preordered sequence (manual)	彼 られ て いる でも 訴え こと 盗ん だ トラム から 停車場 の サウス メルボルン に 金曜日 夜 。
Generated preordered sequence (AWESoME)	彼 いる でも 訴え こと 盗んを トラム から の 停車場 サウス メルボルン に 金曜日 夜。
Baseline translation	He is also accused of stealing a tram from a South Melbourne station on Friday night .
Tagged source-to-target concatenate input by oracle (manual)	He is also accused of stealing a tram on Friday night from a South Melbourne escalator .
Tagged source-to-target concatenate input by oracle (AWESoME)	He is also accused of stealing the tram on Friday night from South Melbourne exit .
Tagged source-to-target concatenate input by mT5 (manual)	He is also accused of stealing a tram from a depot in South Melbourne on Friday night .
Tagged source-to-target concatenate input by mT5 (AWESoME)	He is also accused of stealing the tram from a lane at South Melbourne on Friday night .
Chinese original sequence	此外 , 国外 对 出口 企业 实施 严格 的 责任 标准 。
Japanese target reference sequence	この ほか 、 国際 市場 で は 輸出 企業 に 、 厳格 な 責任 を 課す よう に なっ た 。
	(Additionally, the international market has come to require strict responsibility to exporting companies.)
Oracle (manual) preordered sequence	此外 , 国外 出口 企业 对 严格 的 责任 实施 。
Oracle (AWESoME) preordered sequence	此外 , 国外 出口 企业 对 严 格 的 责任 实施 。
Generated preordered sequence (manual)	此外 , 国外 出口 企业 对 严格 责任 标 准 实施 。
Generated preordered sequence (AWESoME)	此外 , 国外 出口 企业 对 严 格 的 责任 标准 实施 。
Baseline translation	さらに 、 海外 から の 輸出 企業 に 対し て は 厳格 な 責任 基準 が 定め られ て いる 。
Tagged source-to-target concatenate input by oracle (manual)	さらに 、海外 へ の 輸出 企業 に 対し て は 厳格 な 責任管理 を 行っ て いる 。
Tagged source-to-target concatenate input by oracle (AWESoME)	さらに、 海外 へ の 輸出 企業 に 対し て 厳格 な 責任基準 が 課さ れ て いる 。
Tagged source-to-target concatenate input by mT5 (manual)	さらに 、海外 へ の 輸出 企業 に 対し て は 厳格 な 責任基準 を 実施 し て いる 。
Tagged source-to-target concatenate input by mT5 (AWESoME)	さらに、 海外 から の 輸出 企業 に 対し 厳格 な 責任基準 を 課さ れ て いる 。

Table 6: Results of preordering generated by mT5-large.

cides with the preordering performance. The better the preordering quality is, the higher the final translation quality is.

7.3 Specific Results

We have included our experimental results in Table 6 to compare the differences between translations by oracle and seq2seq models. In the first example, the meaningful words ' 金曜日 夜 (Friday night)' and 'サウス メルボルン 停車場 (South Melbourne depot)' were generated in the opposite position. This led to errors in the final translation results when evaluated by the BLEU score, although the transposition of the meaningful words in this example did not affect the semantics of the output text. In the second example, the generated preordered Chinese sequence retained more tokens than the oracle sequence. For example, the Chinese token '标准 (standard)' has the same meaning as '基準' in Japanese. The reference abandoned this word because it was unaligned with any token in the Japanese sequence, while the generated preordered sequence retained it. This resulted in a surplus of translation content compared to the reference Japanese. We also observed that when generating a preordered sequence through a preordering model trained with manual word alignment data, the Chinese conjunction token ' 的' is omitted. However, during the translation process, the decoder is able to appropriately incorporate this token (with 'な' in Japanese) based on the surrounding context. Overall, these examples highlight the importance of paying attention to the position or number of tokens in the preordered sequence, as they can have a significant impact on the final translation quality.

8 Conclusion

In this paper, we propose the utilization of seq2seq multilingual pretrained models for preordering and translation. Specifically, we use manual and mBERT-based word alignment to train mT5-large in generating preordering sequences, and mBART for performing translation. We compare the translation accuracy under various training/test patterns during translation. Our approach is evaluated on ALT Ja-En, Ja-Hi pairs, and in-house Zh-Ja pairs. The results indicate that our proposed approach significantly outperformed the baseline model in most translation directions of Ja-En, Ja-Zh, and Ja-Hi pairs in at least one of BLEU/COMET scores. In future work, we will further explore which kind of input aspect is the most impactful for improving translation tasks.

multilingual-t5 and (Xue et al., 2021).

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Dou, Z.-Y. and Neubig, G. (2021). Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. (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.

Kawara, Y., Chu, C., and Arase, Y. (2018). Recursive neural network based preordering for English-to-Japanese machine translation. In *Proceedings of ACL 2018, Student Research Workshop*, pages 21–27.

Lee, E.-S., Thillainathan, S., Nayak, S., Ranathunga, S., Adelani, D., Su, R., and McCarthy, A. (2022). Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 58–67.

Lin, Z., Pan, X., Wang, M., Qiu, X., Feng, J., Zhou, H., and Li, L. (2020). Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Murthy, R., Kunchukuttan, A., and Bhattacharyya, P. (2019). Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. 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 3868–3873.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Riza, H., Purwoadi, M., Gunarso, Uliniansyah, T., Ti, A. A., Aljunied, S. M., Mai, L. C., Thang, V. T., Thai, N. P., Chea, V., Sun, R., Sam, S., Seng, S., Soe, K. M., Nwet, K. T., Utiyama, M., and Ding, C.

(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.

Sen, S., Gupta, K. K., Ekbal, A., and Bhattacharyya, P. (2019). Multilingual unsupervised NMT using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3083–3089.

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221.

Shi, T., Keneshloo, Y., Ramakrishnan, N., and Reddy, C. K. (2021). Neural abstractive text summarization with sequence-to-sequence models. *ACM/IMS Trans. Data Sci.*, 2(1).

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., and Li, X. (2016). Neural generative question answering. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 36–42.

Zhu, J., Wei, Y., Tamura, T., Utsuro, T., and Nagata, M. (2022). A framework for low resource language translation based on SMT and highly accurate word alignment. In *Proceedings of the 28th Annual Conference of the Association for Natural Language Processing*, pages 1312–1316.