A Bidirectional Transformer Based Alignment Model for Unsupervised Word Alignment

Jingyi Zhang[1] **and Josef van Genabith**[1,2]
[1]German Research Center for Artificial Intelligence (DFKI),
Saarland Informatics Campus, Saarbrücken, Germany
[2]Department of Language Science and Technology, Saarland University,
Saarland Informatics Campus, Saarbrücken, Germany
Jingyi.Zhang@dfki.de,Josef.Van_Genabith@dfki.de

Abstract

Word alignment and machine translation are two closely related tasks. Neural translation models, such as RNN-based and Transformer models, employ a target-to-source attention mechanism which can provide rough word alignments, but with a rather low accuracy. High-quality word alignment can help neural machine translation in many different ways, such as missing word detection, annotation transfer and lexicon injection. Existing methods for learning word alignment include statistical word aligners (e.g. GIZA++) and recently neural word alignment models. This paper presents a bidirectional Transformer based alignment (BTBA) model for unsupervised learning of the word alignment task. Our BTBA model predicts the current target word by attending the source context and both left-side and right-side target context to produce accurate target-to-source attention (alignment). We further fine-tune the target-to-source attention in the BTBA model to obtain better alignments using a full context based optimization method and self-supervised training. We test our method on three word alignment tasks and show that our method outperforms both previous neural word alignment approaches and the popular statistical word aligner GIZA++.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2014; Vaswani et al., 2017) achieves state-of-the-art results for various translation tasks (Barrault et al., 2019, 2020). Neural translation models, such as RNN-based (Bahdanau et al., 2014) and Transformer (Vaswani et al., 2017) models, generally have an encoder-decoder structure with a target-to-source attention mechanism. The target-to-source attention in NMT can provide rough word alignments but with a rather low accuracy (Koehn and Knowles, 2017). High-quality word alignment can be used to help NMT in many different ways, such as detecting source words that are missing in the translation (Lei et al., 2019), integrating an external lexicon into NMT to improve translation for domain-specific terminology or low-frequency words (Chatterjee et al., 2017; Chen et al., 2020), transferring word-level annotations (e.g. underline and hyperlink) from source to target for document/webpage translation (Müller, 2017).

A number of approaches have been proposed to learn the word alignment task, including both statistical models (Brown et al., 1993) and recently neural models (Zenkel et al., 2019; Garg et al., 2019; Zenkel et al., 2020; Chen et al., 2020; Stengel-Eskin et al., 2019; Nagata et al., 2020). The popular word alignment tool GIZA++ (Och and Ney, 2003) is based on statistical IBM models (Brown et al., 1993) which learn the word alignment task through unsupervised learning and do not require gold alignments from humans as training data. As deep neural networks have been successfully applied to many natural language processing (NLP) tasks, neural word alignment approaches have developed rapidly and outperformed statistical word aligners (Zenkel et al., 2020; Garg et al., 2019). Neural word alignment approaches include both supervised and unsupervised approaches: supervised approaches (Stengel-Eskin et al., 2019; Nagata et al., 2020) use gold alignments from human annotators as training data and train neural models to learn word alignment through supervised learning; unsupervised approaches do not use gold human alignments for model training and mainly focus on improving the target-to-source attention in NMT models to produce better word alignment, such as performing attention optimization during inference (Zenkel et al., 2019), encouraging contiguous alignment connections (Zenkel et al., 2020) or using alignments from GIZA++ to supervise/guide the attention in NMT models (Garg et al., 2019).

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 283–292
August 1–6, 2021. ©2021 Association for Computational Linguistics

We propose a bidirectional Transformer based alignment (BTBA) model for unsupervised learning of the word alignment task. Our BTBA model predicts the current target word by paying attention to the source context and both left-side and right-side target context to produce accurate target-to-source attention (alignment). Compared to the original Transformer translation model (Vaswani et al., 2017) which computes target-to-source attention based on only the left-side target context due to left-to-right autoregressive decoding, our BTBA model can exploit both left-side and right-side target context to compute more accurate target-to-source attention (alignment). We further fine-tune the BTBA model to produce better alignments using a full context based optimization method and self-supervised training. We test our method on three word alignment tasks and show that our method outperforms previous neural word alignment approaches and also beats the popular statistical word aligner GIZA++.

2 Background

2.1 Word Alignment Task

The goal of the word alignment task (Och and Ney, 2003) is to find word-level alignments for parallel source and target sentences. Given a source sentence $s_0^{I-1} = s_0, ..., s_i, ..., s_{I-1}$ and its parallel target sentence $t_0^{J-1} = t_0, ..., t_j, ..., t_{J-1}$, the word alignment G is defined as a set of links that link the corresponding source and target words as shown in Equation 1.

$$G \subseteq \{(i, j) : i = 0, ..., I-1; j = 0, ..., J-1\} \quad (1)$$

The word alignment G allows one-to-one, one-to-many, many-to-one, many-to-many alignments and also unaligned words (Och and Ney, 2003). Due to the lack of labelled training data (gold alignments annotated by humans) for the word alignment task, most word alignment methods learn the word alignment task through unsupervised learning (Brown et al., 1993; Zenkel et al., 2020; Chen et al., 2020).

2.2 Neural Machine Translation

Neural translation models (Bahdanau et al., 2014; Vaswani et al., 2017) generally have an encoder-decoder structure with a target-to-source attention mechanism: the encoder encodes the source sentence; the decoder generates the target sentence by attending the source context and performing left-to-right autoregressive decoding. The target-to-source attention learned in NMT models can provide rough word alignments between source and target words. Among various translation models, the Transformer translation model (Vaswani et al., 2017) achieves state-of-the-art results on various translation tasks and is based solely on attention: source-to-source attention in the encoder; target-to-target and target-to-source attention in the decoder. The attention networks used in the Transformer model are called multi-head attention which performs attention using multiple heads as shown in Equation 2.

$$\begin{aligned}
&MultiHead\,(Q, K, V) \\
&\quad = Concat\,(head_0, ..., head_{N-1})\,W^o \\
&Head_n = A_n \cdot V_n \\
&A_n = softmax\left(\frac{Q_n K_n^T}{\sqrt{d_k}}\right) \\
&Q_n = QW_n^Q, K_n = KW_n^K, V_n = VW_n^V
\end{aligned} \quad (2)$$

where Q, K and V are query, keys, values for the attention function; W^o, W_n^Q, W_n^K and W_n^V are model parameters; d_k is the dimension of the keys. Based on parallelizable attention networks, the Transformer can be trained much faster than RNN-based translation models (Bahdanau et al., 2014).

3 Related Work

3.1 Statistical Alignment Models

Word alignment is a key component in traditional statistical machine translation (SMT), such as phrase-based SMT (Koehn et al., 2003) which extracts phrase-based translation rules based on word alignments. The popular statistical word alignment tool GIZA++ (Och and Ney, 2003) implements the statistical IBM models (Brown et al., 1993). The statistical IBM models are mainly based on lexical translation probabilities. Words that co-occur frequently in parallel sentences generally have higher lexical translation probabilities and are more likely to be aligned. The statistical IBM models are trained using parallel sentence pairs with no word-level alignment annotations and therefore learn the word alignment task through unsupervised learning. Based on a reparameterization of IBM Model 2, Dyer et al. (2013) presented another popular statistical word alignment tool fast_align which can be trained faster than GIZA++, but GIZA++ generally produces better word alignments than fast_align.

3.2 Neural Alignment Models

With neural networks being successfully applied to many NLP tasks, neural word alignment approaches have received much attention. The first neural word alignment models are based on feedforward neural networks (Yang et al., 2013) and recurrent neural networks (Tamura et al., 2014) which can be trained in an unsupervised manner by noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010) or in a supervised manner by using alignments from human annotators or existing word aligners as labelled training data.

As NMT (Bahdanau et al., 2014; Vaswani et al., 2017) achieves great success, the target-to-source attention in NMT models can be used to infer rough word alignments, but with a rather low accuracy. A number of recent works focus on improving the target-to-source attention in NMT to produce better word alignments (Garg et al., 2019; Zenkel et al., 2019; Chen et al., 2020; Zenkel et al., 2020). Garg et al. (2019) trained the Transformer translation model to jointly learn translation and word alignment through multi-task learning using word alignments from existing word aligners such as GIZA++ as labelled training data. Chen et al. (2020) proposed a method to infer more accurate word alignments from the Transformer translation model by choosing the appropriate decoding step and layer for word alignment inference. Zenkel et al. (2019) proposed an alignment layer for the Transformer translation model and they only used the output of the alignment layer for target word prediction which forces the alignment layer to produce better alignment (attention). Zenkel et al. (2019) also proposed an attention optimization method which directly optimizes the attention for the test set to produce better alignment. Zenkel et al. (2020) proposed to improve the attention in NMT by using a contiguity loss to encourage contiguous alignment connections and performing direct attention optimization to maximize the translation probability for both the source-to-target and target-to-source translation models. Compared to these methods that infer word alignments based on NMT target-to-source attention which is computed by considering only the left-side target context, our BTBA model can exploit both left-side and right-side target context to compute better target-to-source attention (alignment).

There are also a number of supervised neural approaches that require gold alignments from humans for learning the word alignment task (Stengel-Eskin et al., 2019; Nagata et al., 2020). Because gold alignments from humans are scarce, Stengel-Eskin et al. (2019); Nagata et al. (2020)'s models only have a small size of task-specific training data and exploit representations from pre-trained NMT and BERT models. Compared to these supervised methods, our method does not require gold human alignments for model training.

4 Our Approach

We present a bidirectional Transformer based alignment (BTBA) model for unsupervised learning of the word alignment task. Motivated by BERT which learns a masked language model (Devlin et al., 2019), we randomly mask 10% of the words in the target sentence and then train our BTBA model to predict the masked target words by paying attention to the source context and both left-side and right-side target context. Therefore, our BTBA model can exploit both left-side and right-side target context to compute more accurate target-to-source attention (alignment) compared to the original Transformer translation model (Vaswani et al., 2017) which computes the target-to-source attention based on only the left-side target context due to left-to-right autoregressive decoding. We further fine-tune the target-to-source attention in the BTBA model to produce better alignments using a full context based optimization method and self-supervised training.

4.1 Bidirectional Transformer Based Alignment (BTBA)

Figure 1 shows the architecture of the proposed BTBA model. The encoder is used to encode the source sentence[1] and has the same structure as the original Transformer encoder (Vaswani et al., 2017). The input of the decoder is the masked target sentence and 10% of the words in the target sentence are randomly masked[2]. As shown in Figure 1, the target sentence contains a masked word $<x>$. The decoder contains 6 layers. Each of the first 5 layers of the decoder has 3 sub-layers:

[1] Following Och and Ney (2003)'s work, we add a $<bos>$ token at the beginning of the source sentence for target words that are not aligned with any source words.

[2] During training, we randomly mask 10% of the words in the target sentences for each training epoch, i.e., one target sentence is masked differently for different training epochs. If a target sentence contains less than 10 words, then we just randomly mask one word in this sentence.

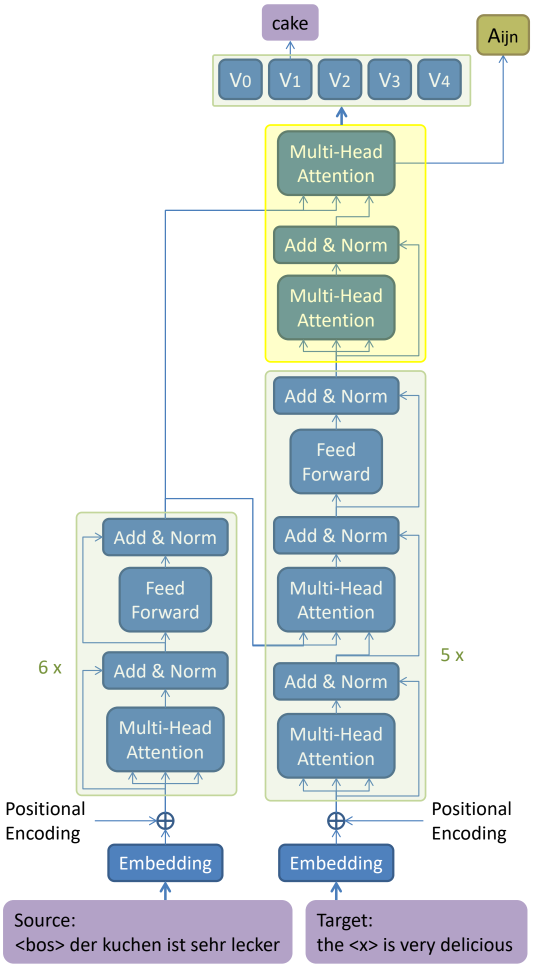

Figure 1: Architecture of our BTBA model.

a multi-head self-attention sub-layer, a target-to-source multi-head attention sub-layer and a feed forward sub-layer, like a standard Transformer decoder layer except that the self-attention sub-layer in the standard Transformer decoder can only attend left-side target context while the self-attention sub-layer in our BTBA decoder can attend all target words and make use of both left-side and right-side target context to compute better target-to-source attention (alignment). The last layer of the BTBA decoder contains a self-attention sub-layer and a target-to-source attention sub-layer like the first 5 layers of the BTBA decoder but without the feed-forward sub-layer. We use the output of the last target-to-source attention sub-layer for predicting the masked target words and we use the attention of the last target-to-source attention sub-layer for inferring word alignments between source and target words. Our design that only uses the last target-to-source attention sub-layer output for predicting the masked target words is motivated by the alignment layer of Zenkel et al. (2019) in order to force the last target-to-source attention sub-layer to pay attention to the most important source words for predicting the target word and therefore produce better word alignments.

Original	the cake is very delicious
Masked	$<x>$ cake is very delicious the $<x>$ is very delicious the cake $<x>$ very delicious the cake is $<x>$ delicious the cake is very $<x>$

Table 1: Masking target sentences in the test set.

In Figure 1, A_{ijn} is the attention value of the jth target word paying to the ith source word using the nth head in the last target-to-source multi-head attention sub-layer. V_0, V_1, V_2, V_3, V_4 are the outputs of the decoder for the 5 target words and V_1 is used to predict the masked target word “cake”. Because V_1 is used to predict “cake”, the attention value A_{21n} should be learned to be high in order to make V_1 contain the most useful source information (“kuchen”). Therefore, A_{ijn} can be used to infer word alignment for the target word “cake” effectively. However, A_{ijn} cannot provide good word alignments for unmasked target words such as “delicious” in Figure 1 because V_4 is not used to predict any target word and A_{54n} is not necessarily learned to be high.

Because A_{ijn} can only be used to infer accurate word alignment for masked target words but we want to get alignments for all target words in the test set, we mask a target sentence t_0^{J-1} in the test set J times and each time we mask one target word as shown in Table 1. Each masked target sentence is fed into the BTBA model together with the source sentence and then we collect the attention A_{ijn} for the masked target words. Suppose the j'th target word is masked, then we compute the source position that it should be aligned to as,

$$i' = \arg\max_{i} \sum_{n=0}^{N-1} A_{ij'n} \quad (3)$$

4.2 Full Context Based Optimization

In Equation 3, the attention $A_{ij'n}$ for the j' target word is computed by considering both left-side and right-side target context, but information about the current target word is not used since the j' target word is masked. For example in Figure 1, the BTBA model does not know that the second target word is “cake” because it is masked, therefore the BTBA model computes the attention (alignment) for “cake” only using the left-side and right-side context of “cake” without knowing that the word that needs to be aligned is “cake”. We propose a novel full context based optimization method to use full target context, including the current target word information, to improve the target-to-source attention in the BTBA model to produce better alignments. That is for the last 50 training steps of the BTBA model, we do not mask the target sentence any more and we only optimize parameters W_n^Q and W_n^K in the last target-to-source multi-head attention sub-layer. As shown in Equation 2, W_n^Q and W_n^K are parameters that are used to compute the attention values in multi-head attention. Optimizing W_n^Q and W_n^K based on full target context can help the BTBA model to produce better attention (alignment) while at the same time freezing other parameters can make the BTBA model keep the knowledge learned from masked target word prediction. After full target context based optimization, we do not need to mask target sentences in the test set as shown in Table 1 any more. We can directly feed the original source and target test sentences into the BTBA model and compute attention (alignment) for all target words in the sentence. The full context based optimization method can be seen as a fine-tuning of the original BTBA model, i.e. we fine-tune the two parameters W_n^Q and W_n^K in the last target-to-source attention layer based on full target context to compute more accurate word alignments.

4.3 Self-Supervised Training

The BTBA model learns word alignment through unsupervised learning and does not require labelled data for the word alignment task. We train two unsupervised BTBA models, one for the forward direction (source-to-target) and one for the backward direction (target-to-source), and then symmetrize the alignments using heuristics such as *grow-diagonal-final-and* (Och and Ney, 2003) as the symmetrized alignments have better quality than the alignments from a single forward or backward model. After unsupervised learning, we use the symmetrized word alignments G_a inferred from our unsupervised BTBA models as labelled data to further fine-tune each BTBA model for the word alignment task through supervised training using the alignment loss in Equation 4 following Garg et al. (2019)’s work.[3] During supervised training, the BTBA model is trained to learn the alignment task instead of masked target word prediction, therefore the target sentence does not need to be masked.

$$L_a(A) = -\frac{1}{|G_a|} \sum_{(p,q)\in G_a} \sum_{n=0}^{N-1} \log(A_{pqn}) \qquad (4)$$

Note that we apply byte pair encoding (BPE) (Sennrich et al., 2016) for both source and target sentences before we feed them into the BTBA model. Therefore the alignments inferred from the BTBA model is on BPE-level. We convert[4] BPE-level alignments to word-level alignments before we perform alignment symmetrization. After alignment symmetrization, we want to use the symmetrized alignments to further fine-tune each BTBA model through supervised learning and therefore we convert[5] the word-level alignments back to BPE-level for supervised training of the BTBA models.

5 Experiments

5.1 Settings

In order to compare with previous work, we used the same datesets[6] as Zenkel et al. (2020)’s work and conducted word alignment experiments for three language pairs: German ↔ English (DeEn), English ↔ French (EnFr) and Romanian ↔ English (RoEn). Each language pair contains a test set and a training set: the test set contains parallel sentences with gold word alignments annotated by humans; the training set contains only parallel sentences with no word alignments. Table 2 gives numbers of sentence pairs contained in the training and test sets. Parallel sentences from both the training set and the test set can be used to train

[3] We optimize all model parameters during supervised fine-tuning.

[4] To convert BPE-level alignments to word-level alignments, we add an alignment between a source word and a target word if any parts of these two words are aligned. Alignments between the source <bos> token and any target word are deleted; alignments between the last source word “.” (full stop) and a target word which is not the last target word are also deleted.

[5] To convert word-level alignments to BPE-level alignments, we add an alignment between a source BPE token and a target BPE token if the source word and the target word that contain these two BPE tokens are aligned; we add an alignment between the source <bos> token and a target BPE token if the target word that contains this target BPE token is not aligned with any source words.

[6] https://github.com/lilt/alignment-scripts

	DeEn	EnFr	RoEn
TRAIN	1.91M	1.13M	447k
TEST	508	447	248

Table 2: Numbers of sentence pairs in the datasets.

Method		DeEn	EnFr	RoEn
Zenkel et al. (2019)		21.2%	10.0%	27.6%
Garg et al. (2019)		16.0%	4.6%	23.1%
Zenkel et al. (2020)		16.3%	5.0%	23.4%
Chen et al. (2020)		15.4%	4.7%	21.2%
GIZA++		18.4%	5.2%	24.2%
Ours	BTBA-left	30.3%	20.2%	33.0%
	BTBA-right	32.3%	14.9%	38.6%
	BTBA	17.8%	9.5%	22.9%
	+ FCBO	16.3%	8.9%	20.6%
	+ SST	**14.3%**	6.7%	**18.5%**
	+ GST	14.5%	**4.2%**	19.7%

Table 3: AER Results. FCBO: full context based optimization; SST: self-supervised training; GST: GIZA++ supervised training.

unsupervised word alignment models. We use BPE (Sennrich et al., 2016) to learn a joint source and target vocabulary of 40k. After BPE, we train BTBA models to learn the word alignment tasks. We use a word embedding size of 512. The feed forward layer contains 2048 hidden units. The multi-head attention layer contains 8 heads. We use the Adam (Kingma and Ba, 2014) algorithm for optimization and set the learning rate to 0.0002. We use a dropout of 0.3. Each training batch contains 40k masked target words. Since the word alignment tasks do not provide validation data, we trained all BTBA models for a fixed number of training epochs: 50 for DeEn, 100 for EnFr and 200 for RoEn.[7] For the last 50 training steps of each BTBA model, we performed full context based optimization.

For each language pair, we trained two BTBA models, one for the forward direction and one for the backward direction, and then symmetrized the alignments. We tested different heuristics for alignment symmetrization, including the standard Moses heuristics, *grow-diagonal*, *grow-diagonal-final*, *grow-diagonal-final-and*. We also tested another heuristic *grow-diagonal-and* which is slightly different from *grow-diagonal*: the *grow-diagonal-and* heuristic only adds a new alignment (i, j) when both s_i and t_j are unaligned while *grow-diagonal* adds a new alignment (i, j) when any of the two words (s_i and t_j) are unaligned. We find that the Moses heuristic *grow-diagonal-final-and* generally achieved the best results for symmetrizing the BTBA alignments, but *grow-diagonal-and* worked particularly good for the EnFr task.

Finally, we used the symmetrized alignments inferred from our unsupervised BTBA models as labelled data to further fine-tune each BTBA model to learn the alignment task through supervised training. We fine-tuned each BTBA model for 50 training steps using the alignment loss in Equation 4. In addition, we also tested to use alignments from GIZA++ instead of alignments inferred from our unsupervised BTBA models as labelled data for supervised fine-tuning of the BTBA models.

5.2 Results

Table 3 gives alignment error rate (AER) (Och and Ney, 2000) results of our BTBA model and comparison with previous work. Table 3 also gives results of BTBA-left and BTBA-right: BTBA-left means that the BTBA decoder only attends left-side target context; BTBA-right means that the BTBA decoder only attends right-side target context. As shown in Table 3, the BTBA model, which uses both left-side and right-side target context, significantly outperformed BTBA-left and BTBA-right. Results also show that the performance of our BTBA model can be further improved by full context based optimization (FCBO) and supervised training including both self-supervised training and GIZA++ supervised training. For DeEn and RoEn tasks, the self-supervised BTBA (S-BTBA) model achieved the best results, outperforming previous neural and statistical methods. For the EnFr task, as the statistical aligner GIZA++ performed well and achieved better results than our unsupervised BTBA model, the GIZA++ supervised BTBA (G-BTBA) model achieved better results than the S-BTBA model and also outperformed the original GIZA++ and previous neural models.

Tables 4, 5 and 6 give results of using different heuristics for symmetrizing alignments produced by BTBA, GIZA++ and G-BTBA, respectively. For our unsupervised and self-supervised BTBA models, *grow-diagonal-final-and* achieved the best results on DeEn and RoEn tasks while *grow-diagonal-and* achieved the best results on the EnFr task. For GIZA++ and G-BTBA, the best heuristics for different language pairs are quite different, though *grow-diagonal-final-and* generally

[7] The training time (time of one training epoch × number of training epochs) of one BTBA model for different tasks (DeEn, EnFr and RoEn) is roughly the same, 30 hours using 4 GPUs.

	DeEn			EnFr			RoEn		
	BTBA	+FCBO	+SST	BTBA	+FCBO	+SST	BTBA	+FCBO	+SST
forward	20.2%	18.3%	**14.3%**	13.6%	12.8%	7.3%	24.7%	22.4%	20.5%
backward	23.8%	23.3%	17.2%	14.6%	13.3%	7.5%	27.3%	26.1%	22.0%
union	20.6%	18.3%	14.5%	15.7%	14.3%	7.5%	24.1%	21.2%	18.9%
intersection	23.7%	23.9%	17.1%	11.6%	11.2%	7.4%	28.3%	27.9%	24.0%
grow-diagonal	19.9%	18.5%	**14.3%**	11.2%	10.7%	6.9%	23.6%	21.6%	18.6%
grow-diagonal-and	21.0%	20.6%	17.3%	**9.5%**	**8.9%**	**6.7%**	26.1%	25.4%	23.6%
grow-diagonal-final	19.5%	17.3%	14.4%	14.4%	13.4%	7.4%	23.4%	20.8%	18.6%
grow-diagonal-final-and	**17.8%**	**16.3%**	**14.3%**	11.9%	11.2%	7.0%	**22.9%**	**20.6%**	**18.5%**

Table 4: Comparison of different heuristics for symmetrizing the BTBA alignments. FCBO: full context based optimization. SST: self-supervised training.

	DeEn	EnFr	RoEn
forward	19.0%	10.3%	25.6%
backward	22.5%	9.1%	29.7%
union	22.1%	12.9%	27.5%
intersection	19.0%	**5.2%**	27.8%
grow-diagonal	**18.4%**	7.7%	24.5%
grow-diagonal-and	18.9%	5.7%	26.1%
grow-diagonal-final	21.1%	11.7%	26.0%
grow-diagonal-final-and	18.9%	8.5%	**24.2%**

Table 5: Comparison of different heuristics for symmetrizing GIZA++ alignments.

	DeEn	EnFr	RoEn
forward	**14.5%**	5.8%	21.4%
backward	17.6%	**4.2%**	21.9%
union	15.1%	5.3%	19.9%
intersection	17.2%	4.7%	23.6%
grow-diagonal	14.7%	4.6%	**19.7%**
grow-diagonal-and	17.5%	4.4%	23.7%
grow-diagonal-final	15.1%	5.3%	19.8%
grow-diagonal-final-and	14.8%	4.7%	19.8%

Table 6: Comparison of different heuristics for symmetrizing G-BTBA alignments.

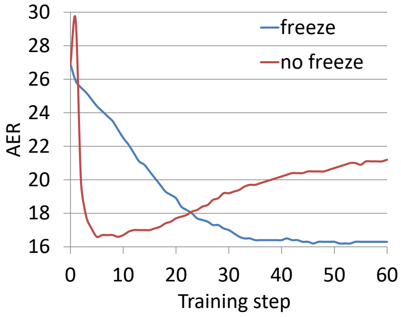

Figure 2: DeEn test AER per training step during FCBO with/without parameter freezing.

obtained good (best or close to best) results on DeEn and RoEn tasks while *grow-diagonal-and* generally obtained good (close to best) results on the EnFr task.

FCBO with/without Parameter Freezing As we explained in Section 4.2, during full context based optimization (FCBO), we only optimize W_n^Q and W_n^K in the last target-to-source attention sub-layer and freeze all other parameters so the BTBA model can keep the knowledge learned from masked target word prediction. We also tested to optimize all parameters of the BTBA model without parameter freezing during FCBO. Figure 2 shows how the AER results on the DeEn test set changed during FCBO with and without parameter freezing. Without freezing any parameters during FCBO, the AER result (the red curve) first increased a little, then decreased sharply, and soon increased again. In contrast, when we freeze most of the parameters, the AER result (the blue curve) decreased stably and eventually got better results (16.3%) than no parameter freezing (16.7%). Note that the results in Figure 2 are computed based on full target context, i.e., the target sentence is not masked. As we explained in Section 4.1, the BTBA model without FCBO should only be used to infer word alignments for masked target words. Without FCBO, using the BTBA model to infer word alignments for unmasked target words produces poor AER results (26.9% as shown in Figure 2) compared to using the BTBA model to infer word alignments for masked target words (17.8% as shown in Table 3). FCBO can quickly improve the results of using the BTBA model for inferring word alignments for unmasked target words, and eventually after FCBO, the BTBA model can effectively use full target context to compute better word alignment compared to the original BTBA model without FCBO (16.3% versus 17.8% as shown in Table 3).

Training Data for Supervised Learning Because the symmetrized BTBA alignments have better quality compared to alignments from a single unidirectional (forward or backward) BTBA model as shown in Table 4, we used the symmetrized

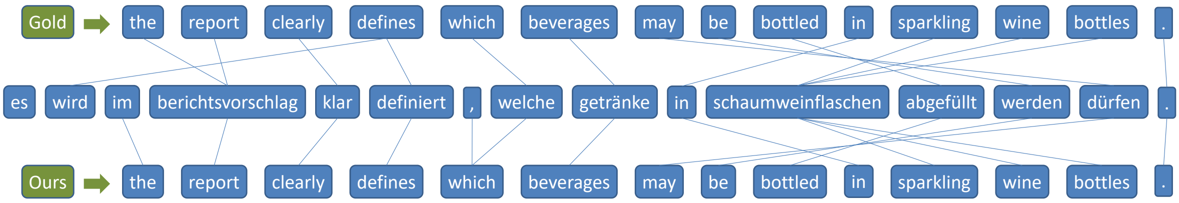
Gold
the report clearly defines which beverages may be bottled in sparkling wine bottles .
es wird im berichtsvorschlag klar definiert , welche getränke in schaumweinflaschen abgefüllt werden dürfen .
Ours
the report clearly defines which beverages may be bottled in sparkling wine bottles .

Figure 3: An example of gold alignments and alignments produced by our S-BTBA model.

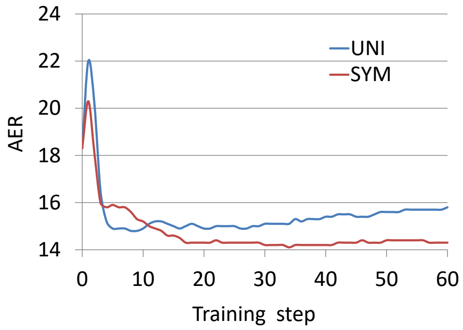

Figure 4: AER results of the forward BTBA model during self-supervised training. UNI: using unidirectional BTBA alignments as labelled training data. SYM: using symmetrized BTBA alignments as labelled training data.

		DeEn	EnFr	RoEn
S-BTBA	FF	12.3	11.3	18.2
	CC	6.1	3.3	7.8
	FC	44.4	12.8	41.1
G-BTBA	FF	13.2	5.1	18.6
	CC	7.1	2.9	8.3
	FC	43.3	9.3	46.1

Table 7: AER for different types of alignments.

word alignments inferred from our unsupervised BTBA models as labelled data to further fine-tune each unidirectional BTBA model for the alignment task through supervised training. We also tested to use unidirectional BTBA alignments instead of symmetrized BTBA alignments as labelled data for supervised training. Figure 4 (the blue curve) shows how the performance of the forward BTBA model of the DeEn task changes during supervised training when using unidirectional alignments inferred from itself (the forward BTBA model) as labelled training data, which demonstrates that the forward BTBA model can be significantly improved through supervised training even when the training data is inferred from itself and not improved by alignment symmetrization. Figure 4 also shows that using symmetrized alignments for supervised training (the red curve) did achieve better results than using unidirectional alignments for supervised training. In addition, it is worth noting that supervised training can improve the BTBA model even if the quality of the labelled training data is somewhat worse than the BTBA model itself, e.g. for the RoEn task, using the GIZA++ alignments for fine-tuning the forward BTBA model through supervised training improved the result of the forward BTBA model (22.4% → 21.4% as shown in Table 4 and Table 6) even though GIZA++ produced worse alignments (24.2% in Table 3) than the forward BTBA model.

Alignment Error Analysis We analyze the alignment errors produced by our system and find that most of the alignment errors are caused by function words. As shown in the alignment example in Figure 3, source and target corresponding content words (e.g. "definiert" and "defines") are all correctly aligned by our model, but function words such as "the", "im" and "wird" are not correctly aligned. To give a more detailed analysis, we compute AER results of our model for 3 different types of alignments: FF (alignments between two function words), CC (alignments between two content words) and FC (alignments between a function word and a content word).[8] Table 7 shows that our models achieved significantly better results for CC alignments than for FF and FC alignments. Function words are more difficult to align than content words most likely because content words in a parallel sentence pair usually have very clear corresponding relations (such as "defines" clearly corresponds to "definiert" in Figure 3), but function words (such as "the", "es" and "im") are used more flexibly and frequently do not have clear corresponding words in parallel sentences, which increases the alignment difficulty significantly.

[8] For each language, we judge whether a word is a function word or a content word using a list of stopwords from nltk, https://www.nltk.org/

	de→en	en→de
SHIFT-AET	34.8	28.0
Ours	35.1	28.7

Table 8: Translation results (BLEU) for dictionary-guided NMT.

5.3 Dictionary-Guided NMT via Word Alignment

For downstream tasks, word alignment can be used to improve dictionary-guided NMT (Song et al., 2020; Chen et al., 2020). Specifically, at each decoding step in NMT, Chen et al. (2020) used a SHIFT-AET method to compute word alignment for the newly generated target word and then revised the newly generated target word by encouraging the pre-specified translation from the dictionary. The SHIFT-AET alignment method adds a separate alignment module to the original Transformer translation model (Vaswani et al., 2017) and trains the separate alignment module using alignments induced from the attention weights of the original Transformer. To test the effectiveness of our alignment method for improving dictionary-guided NMT, we used the alignments inferred from our BTBA models as labelled data for supervising the SHIFT-AET alignment module and performed dictionary-guided translation for the German↔English language pair following Chen et al. (2020)'s work. Table 8 gives the translation results of dictionary-guided NMT and shows that our alignment method led to higher translation quality compared to the original SHIFT-AET method.

6 Conclusion

This paper presents a novel BTBA model for unsupervised learning of the word alignment task. Our BTBA model predicts the current target word by paying attention to the source context and both left-side and right-side target context to produce accurate target-to-source attention (alignment). We further fine-tune the target-to-source attention in the BTBA model to obtain better alignments using a full context based optimization method and self-supervised training. We test our method on three word alignment tasks and show that our method outperforms both previous neural alignment methods and the popular statistical word aligner GIZA++.

Acknowledgments

This work is supported by the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Loc Barrault, Magdalena Biesialska, Ondej Bojar, Marta R. Costa-juss, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubei, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Loc Barrault, Ondej Bojar, Marta R. Costa-juss, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Mller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168, Copenhagen, Denmark. Association for Computational Linguistics.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

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.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Wenqiang Lei, Weiwen Xu, Ai Ti Aw, Yuanxin Xiang, and Tat Seng Chua. 2019. Revisit automatic error detection for wrong and missing translation – a supervised approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 942–952, Hong Kong, China. Association for Computational Linguistics.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Masaaki Nagata, Chousa Katsuki, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual bert. *arXiv preprint arXiv:2004.14516*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8886–8893.

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 910–920, Hong Kong, China. Association for Computational Linguistics.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480, Baltimore, Maryland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 166–175, Sofia, Bulgaria. Association for Computational Linguistics.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.