Explicit Alignment Learning for Neural Machine Translation

Zuchao Li[1,2] , **Hai Zhao**[1,2,*] , **Fengshun Xiao**[1,2] , **Masao Utiyama**[3] and **Eiichiro Sumita**[3]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
{charlee,felixxiao}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, {mutiyama,eiichiro.sumita}@nict.go.jp

Abstract

Even though neural machine translation (NMT) has become the state-of-the-art solution for end-to-end translation, it still suffers from a lack of translation interpretability, which may be conveniently enhanced by explicit alignment learning (EAL), as performed in traditional statistical machine translation (SMT). To provide the benefits of both NMT and SMT, this paper presents a novel model design that enhances NMT with an additional training process for EAL, in addition to the end-to-end translation training. Thus, we propose two approaches an explicit alignment learning approach, in which we further remove the need for the additional alignment model, and perform embedding mixup with the alignment based on encoder–decoder attention weights in the NMT model. We conducted experiments on both small-scale (IWSLT14 De→En and IWSLT13 Fr→En) and large-scale (WMT14 En→De, En→Fr, WMT17 Zh→En) benchmarks. Evaluation results show that our EAL methods significantly outperformed strong baseline methods, which shows the effectiveness of EAL. Further explorations show that the translation improvements are due to a better spatial alignment of the source and target language embeddings. Our method improves translation performance without the need to increase model parameters and training data, which verifies that the idea of incorporating techniques of SMT into NMT is worthwhile.

1 Introduction

Neural machine translation (NMT) has achieved great progress with the development of model structures in an encoder–decoder framework [Bahdanau *et al.*, 2014; Sutskever *et al.*, 2014]. In this framework, various advanced neural architectures have been explored, ranging from recurrent neural networks (RNNs) [Sutskever *et al.*, 2014; Luong *et al.*, 2015] and convolutional neural networks (CNNs) [Gehring *et al.*, 2017a; Gehring *et al.*, 2017b] to full attention networks without recurrence or convolution [Vaswani *et al.*, 2017]. In particular, the self-attention-based Transformer model has achieved state-of-the-art performance for many translation tasks [Vaswani *et al.*, 2017].

Unlike conventional statistical machine translation (SMT) [Koehn *et al.*, 2003b], which explicitly models the word alignment information of the training corpus, NMT uses mechanisms known as attention techniques [Bahdanau *et al.*, 2014] to learn complex alignment relations between source and target sentences in a latent manner. However, evidence has emerged that explicit alignment learning (EAL) may be more helpful. Even though NMT achieves higher translation accuracy than SMT, traditional statistical word-alignment models often outperform NMT models on the closely related task of word alignment [Garg *et al.*, 2019a]. Therefore, it is worth exploring the use of EAL, as in SMT, to help NMT to learn alignment information.

Finding explicit alignment between source and target words has many applications in machine translation, such as generating bilingual lexica from parallel corpora. [Alkhouli and Ney, 2017; Chatterjee *et al.*, 2017; Alkhouli *et al.*, 2018] use word alignments for external-dictionary-assisted translation, to improve translation of low-frequency words or to comply with certain terminology guidelines. There have been a few attempts to adopt SMT alignment or full outputs to aid NMT. Generally, such techniques may be applied in either the training or decoding phase. For the former, previous work has usually used the SMT outcome as an auxiliary feature or component in NMT modeling [Zhou *et al.*, 2017; Zhao *et al.*, 2018], which may not fully exploit the explicit alignment information offered by SMT models. For the latter, the outcome of SMT may be used as hard constraints to guide NMT decoding, which may potentially reduce the speed of translation [Stahlberg *et al.*, 2016].

In this paper, we present an innovative EAL approach to exploit explicit alignment in NMT training. Specifically, to eliminate the main difficulties of most methods that employ SMT to assist NMT, we use the cross-attention in the NMT model as alignment information to complete the replacement or mixup of embedding. Through these approaches, because the source embedding and target embedding of aligned tokens are placed in the same context and optimized with the same target sequence, they are explicitly aligned in the embedding

*Corresponding author. This work was partially funded by the Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

space during the training process. Moreover, the model is driven to discriminate this input with our alignment noise from the true input, thereby achieving the effect of implicit denoising.

With motivation similar to that of our proposed EAL, [Garg *et al.*, 2019b] used discrete alignments from SMT as extra training targets to optimize the cross-attention probabilities, resulting in translation and alignment learning at the same time. The differences are that our EAL applies alignment learning to the embedding space rather than the cross-attention, and our EAL approach totally eliminates the need for an external alignment model, thereby achieving the goal of self-training.

To verify the effectiveness of our proposed methods, we conducted experiments on large-scale benchmarks (WMT14 English-to-German, English-to-French, and WMT17 Chinese-to-English) and low-resource datasets (IWSLT14 German-to-English and IWSLT13 French-to-English). The experimental results show that our method can obtain significant improvement in BLEU score over strong baseline methods, particularly for the low-resource scenarios. Additional exploration shows that the reason for the improvement shown by our proposed EAL methods is that the embedding spaces of the source and target languages are better aligned.

2 Related Work

There are two methods of incorporating SMT into NMT: using the intermediate or final output of SMT as an input to NMT, and applying SMT techniques—such as distortion—to NMT. There have been a few attempts to fuse SMT outputs into NMT. These can be placed into two categories, according to their working phase: training or decoding. [Stahlberg *et al.*, 2016] extended beam-search decoding by expanding the search space of NMT with translation hypotheses produced by a syntactic SMT model. [He *et al.*, 2016] presented a log-linear model to integrate SMT features into NMT. [Mi *et al.*, 2016] and [Liu *et al.*, 2016] proposed a supervised attention model for NMT to minimize the alignment disagreement between NMT and SMT. [Wang *et al.*, 2017] used SMT to offer additional recommendations of generated words according to the decoding information from NMT at each decoding step. [Zhou *et al.*, 2017] proposed a neural combination model to fuse the NMT translation results and SMT translation results. [Zhao *et al.*, 2018] proposed to incorporate a phrase translation table as recommendation memory into NMT systems to alleviate the problem that NMT systems generate fluent but unfaithful translations.

The use of SMT techniques in NMT has also attracted a great deal of interest. Inspired by distortion models [Brown *et al.*, 1993; Koehn *et al.*, 2003a; Tillmann, 2004; Och *et al.*, 2004; Al-Onaizan and Papineni, 2006], which originated from SMT, [Zhang *et al.*, 2017] explicitly incorporated word-reordering knowledge into NMT by forcing the attention mechanism to attend to source words regarding both the semantic requirement and the word-reordering penalty. [Shaw *et al.*, 2018] proposed to encode the order information in a sentence with relative position representations. These approaches implicitly integrate reordering information instead of explicitly reordering the word sequence, as performed by SMT. The decoding is intended to find the best scoring translation from an exponential number of choices in a left-to-right (or right-to-left) manner. Future cost [Koehn *et al.*, 2003a] was proposed in SMT to minimize the risk of pruning the true hypotheses in decoding. Motivated by this idea, [Zheng *et al.*, 2019] adopted a capsule network to model the translated (past) and untranslated (future) contents through parts-to-wholes assignment, which is learned by a routing-by-agreement mechanism. [Duan *et al.*, 2021] proposed to adopt an additional mechanism to predict target words to estimate the future cost during training and decoding. As in the alignment of SMT, [Zhang *et al.*, 2021] constructed a search space similar to that of phrase-based SMT for an NMT model, whereby explicit phrase alignment is readily introduced into the translation process of an arbitrary NMT model. [Garg *et al.*, 2019b] presented an alignment training objective to train a Transformer model to produce both accurate alignments in the cross-attention component. [Song *et al.*, 2020] introduced a dedicated head in the multi-head Transformer architecture to capture external alignment supervision signals. For data-augmentation, [Wang *et al.*, 2018] examined a simple strategy that randomly replace words in both the source sentence and the target sentence with other random words. Our EAL method is similar to these methods in motivation, but has the following differences. 1) Our EAL does not need to make any changes to the model structure or increase the number of parameters of the model, but instead reuses the existing structures. 2) We do not need to introduce any new loss function; alignment and translation are performed by cross-entropy loss, which unifies our alignment training process and translation training process, and saves training time. 3) Our EAL can be self-trained, without relying on external alignment models.

3 Our Approach

We first describe the background and then our proposed EAL method. The intuition behind our approach is that a word and its corresponding word in the target language should have similar embeddings because the source language and target language embedding spaces are aligned in the training. We can replace words with their aligned counterparts to add noise to the source sentences while maintaining semantic closeness.

3.1 Background

The NMT training phase is given a source and target sentence pair $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} = (x_1, x_2, \cdots, x_M)$ is a source-language sentence and $\mathbf{Y} = (y_1, y_2, \cdots, y_N)$ is a target-language sentence. A typical NMT system models the conditional probability according to an encoder–decoder framework with an attention mechanism [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2014]:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{N} P(y_j|\mathbf{Y}_{<j}, \mathbf{X}, \theta), \tag{1}$$

where θ denotes the model parameters and $\mathbf{Y}_{<j}$ denotes the sequence of tokens generated before timestep j.

The encoder and decoder can be specialized using various neural architectures, including gated recurrent units (GRUs) [Bahdanau *et al.*, 2014], long short-term memory (LSTM), CNNs [Gehring *et al.*, 2017b], and Transformer [Vaswani *et al.*, 2017], among which the self-attention-based Transformer is the state-of-the-art architecture for NMT.

The decoder predicts the corresponding translation targets $\mathbf{Y} = (y_1, \cdots, y_N)$, step by step, according to the last decoding state and source context. The translation probability can be formulated as follows:

$$P(y_j|\mathbf{Y}_{<j}, \mathbf{X}) = q(y_{j-1}, s_j, c_j) \tag{2}$$

where s_j and c_j denote the decoding state and source context, respectively, at the j-th timestep, and $q(\cdot)$ is the softmax layer. Specifically,

$$s_j = g(y_{j-1}, s_{j-1}, c_j) \tag{3}$$

where $g(\cdot)$ is the corresponding neural architecture unit. The context vector c_j is defined as the weighted sum of the source representations $h_i^s \in H^s$ on the basis of the cross-attention mechanism:

$$c_j = \sum_{i=1}^{M} \alpha_{ji} h_i^s \tag{4}$$

The alignment component α_{ji} measures the similarity between s_j and h_i. The whole model is jointly trained to seek the optimal parameters that can be used to correctly encode the source sentences and decode them to corresponding target sentences. The final loss for the model optimization is

$$\mathcal{L}(\mathbf{X}, \mathbf{Y}) = -\sum_{j=1}^{N} logP(y_j|\mathbf{Y}_{<j}, \mathbf{X}). \tag{5}$$

In the actual implementation, we chose the state-of-the-art Transformer structure as the baseline, so we further introduce the background of the Transformer NMT model. However, our method is general and can be adapted to other NMT baselines. The Transformer-based NMT model employs a self-attention network for both the encoder and decoder. The encoder and decoder are composed of L_E and L_D stacked multi-head self-attention (MHA) layers, respectively. For an MHA layer in the encoder,

$$\begin{aligned} head_k &= \text{Attn}(H) = \sigma(QW^Q, KW^K, VW^V)W^O \\ \text{MHA}(H) &= \text{Concat}(head_1, \cdots, head_{\mathcal{K}})W^O, \end{aligned}$$

where $Q = \text{Linear}_Q(H)$, $K = \text{Linear}_K(H)$, $V = \text{Linear}_V(H)$, and W^O, W^Q, W^K, and W^V are projection parameters. The self-attention operation σ is the dot-product between key, query, and value pairs:

$$\sigma(Q_1, K_1, V_1) = \text{Softmax}(\frac{Q_1 K_1^T}{\sqrt{d_k}})V_1,$$

where $d_k = d_{model}/\mathcal{K}$ is the dimension of each head. In the decoder layer, in addition to the Self-MHA structure, a Cross-MHA is used to compute the alignment score α_{ji}:

$$\begin{aligned} H^s &= \text{Self-MHA}^s(\mathbf{Emb}(\mathbf{X}) + \mathbf{Pos}(\mathbf{X})), \\ H^t &= \text{Self-MHA}^t(\text{IncMask}(\mathbf{Emb}(\mathbf{Y}) + \mathbf{Pos}(\mathbf{Y}))), \\ \alpha_{ji} &= \text{Cross-MHA}(H^t, H^s). \end{aligned}$$

3.2 Alignment

An alignment $\mathcal{A}$ in machine translation can be defined as a subset of the Cartesian product of the token positions in a source and target sentence pair $(\mathbf{X}, \mathbf{Y})$:

$$\mathcal{A} \subseteq \{(i, j) : i = 1, \cdots, M; j = 1, \cdots, N\},$$

The alignment is intended to be a discrete pair representing a many-to-many mapping from the source tokens to their corresponding translations in the target sentence.

Latently, NMT models may learn the alignment between source token x_i and target token y_j according to two main aspects of X: cross-attention and token embeddings. Because attention weight α_{ji} measures the similarity between h_j^t and h_i^s, it has been widely used to evaluate the token alignment between y_j and x_i, so that the token alignment is explicitly modeled.

In unsupervised NMT models, token alignment learning is performed by updating token embeddings when training. In monolingual vector space, similar words tend to have commonalities in the same dimensions of their token vectors [Mikolov *et al.*, 2013]. These commonalities include (1) a similar degree (value) of the same dimension and (2) a similar positive or negative correlation of the same dimension. In bilingual vector space, [Liu *et al.*, 2019] assumed that the source and target words that have similar meanings should also have similar embedding vectors. Hence, they proposed to perform a sharing technique between source and target token embedding space, resulting in a significant improvement in alignment quality and translation performance. In this work, our method also uses this premise to optimize the embedding of the aligned words that have the same context, thereby driving the model to learn the alignment of source–target embeddings, instead of relying on the model itself to implicitly perform alignment learning with the cross-attention.

3.3 Explicit Alignment Learning

In our proposed EAL approach, we employ the cross-attention in the decoder to provide alignment weights and obtain the target-language representation of the source positions, and then mixup with the original source input embedding to achieve the alignment learning. The overall architecture is shown in Fig. 1.

The entire EAL architecture consists of two processes: the ordinary end-to-end translation learning process and the alignment learning process. In the end-to-end translation learning process, as introduced in Section 3.1, we calculate H^s and H^t, and then use cross-attention to calculate the alignment weight of $src \rightarrow tgt$:

$$\begin{aligned} a_{src \rightarrow tgt} &= H^t \times (H^s)^T \\ \alpha_{src \rightarrow tgt} &= \text{Softmax}\big(a_{src \rightarrow tgt}\big), \end{aligned} \tag{6}$$

where H^s has shape $B \times M \times d$, H^t has shape $B \times N \times d$, $src \rightarrow tgt$ means that the alignment weight is calculated with target representation tgt as the query and source representation src as the key, and the obtained alignment weight has shape $B \times N \times M$. The purpose of $\alpha_{tgt \rightarrow src}$ is to map the representation of the source sequence to the target positions,

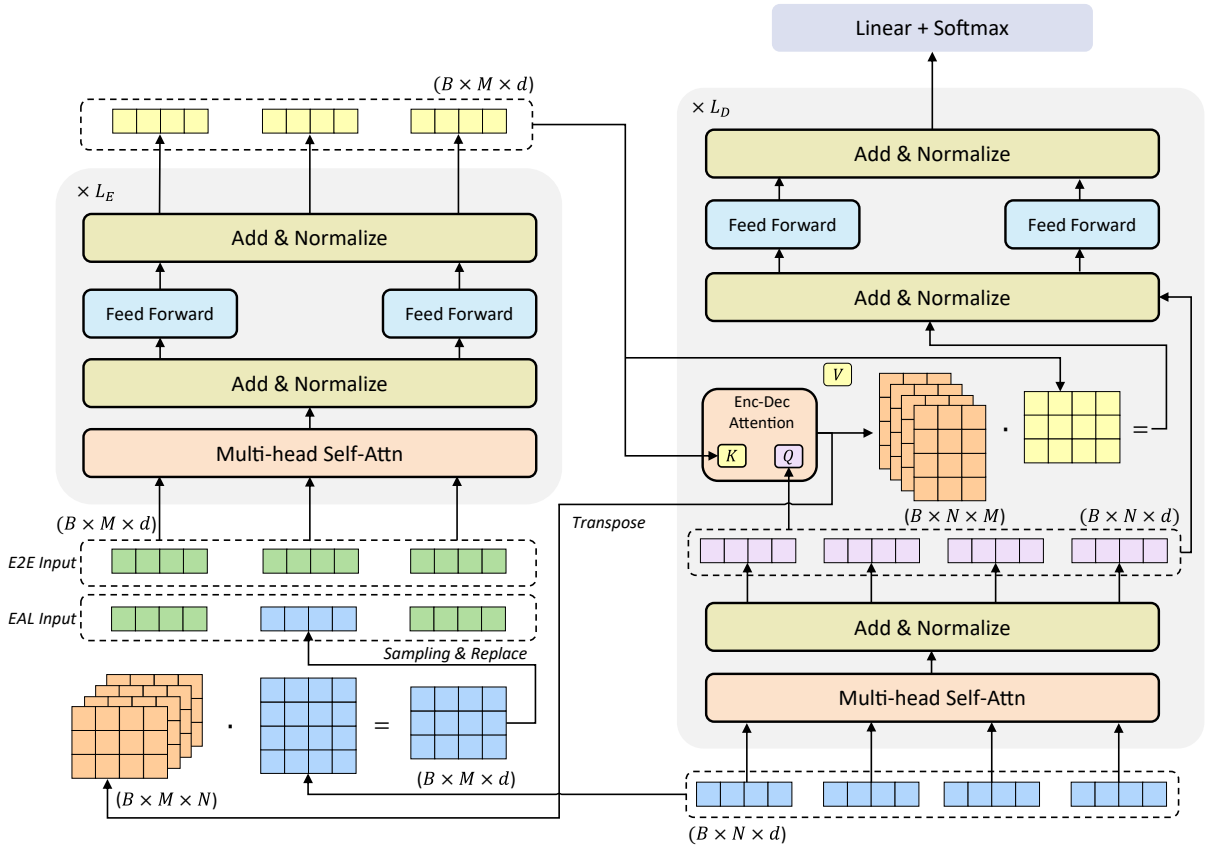

Figure 1: Architecture of our proposed model.

and Softmax is performed on the source sequence dimension M.

In our EAL approach, the goal is to place the embedding of the target at its aligned source position, which is contrary to the alignment weights obtained by cross-attention in the decoder, that is, what is needed in EAL is $tgt \rightarrow src$. We also explain how to calculate $tgt \rightarrow src$. According to the matrix operation rules, we found that

$$\begin{aligned} a_{tgt \rightarrow src} &= H^s \times (H^t)^T = \left(H^t \times (H^s)^T\right)^T \\ &= \left(a_{src \rightarrow tgt}\right)^T, \\ \alpha_{tgt \rightarrow src} &= \text{Softmax}\left(a_{tgt \rightarrow src}\right) \\ &= \text{Softmax}\left(\left(a_{src \rightarrow tgt}\right)^T\right), \end{aligned} \tag{7}$$

which means that the weight matrix $a_{tgt \rightarrow src}$ can be reused in calculating $\alpha_{tgt \rightarrow src}$ and $\alpha_{tgt \rightarrow src}$. The only difference is that Softmax in $\alpha_{tgt \rightarrow src}$ is performed on the target sequence dimension N.

Since there is an alignment weight between any two tokens, we can use this weight to convert the target embedding to various positions of the source input, and then use the mixup strategy to obtain a new input representation:

$$\mathbf{E}(x_i) = (1 - \gamma_2)\mathbf{Emb}(x_i) + \gamma_2 \alpha_{tgt \rightarrow src}(x_i)\mathbf{Emb}(\mathbf{Y}). \tag{8}$$

In fact, as shown in Fig. 1, we adopt a sampling process that only use the representation from the target alignment calculation to randomly and partially replace the original input embedding, rather than all or too many ratios. On the one hand, this prevent excessive replacement of missing too much source input information, and on the other hand, through random sampling, each training of the same instance will be replaced in different places, reducing the risk of overfitting.

In general, the process of training the model is shown in Algorithm 1. Our proposed EAL method completely removes the requirement for the external alignment model, and naturally uses the internal alignment weight, which makes our method a self-training method of alignment learning. That is, better internal cross-attention alignment weights result in a better embedding spatial structure alignment, thereby improving the cross-attention alignment. Because it is a self-training method, our training also needs to be divided into two sub-phases. The first phase is the warmup phase, for end-to-end translation training only: to train cross-attention to avoid poor cross-attention alignment in the initial training stage. The second phase is the self-training stage of EAL, in which translation and EAL training are performed at the same time, so that the model achieves better performance. We set the number of warmup steps to w.

In terms of implementation, because the decoder is formed by stacking the same structure of the L_D layers, and the Cross-MHA in each decoder layer is also a multi-head atten-

Algorithm 1: Explicit Alignment Learning

for $t = 1, 2, \cdots, N$ **do**
  $H^s \leftarrow \text{Self-MHA}^s(\mathbf{Emb}(\mathbf{X}) + \mathbf{Pos}(\mathbf{X}))$;
  $H^t \leftarrow \text{Self-MHA}^t(\text{IncMask}(\mathbf{Emb}(\mathbf{Y}) + \mathbf{Pos}(\mathbf{Y})))$;
  $a_{src \to tgt} \leftarrow \text{Cross-MHA}(H^t, H^s)$;
  End2end Translation Training
    $\alpha_{src \to tgt} \leftarrow \text{Softmax}(a_{src \to tgt})$;
    $\mathcal{L}_{\text{E2E}} \leftarrow \text{Softmax}(\text{Linear}(H^t + \alpha_{src \to tgt} H^s))$;
  EAL Training
    $\alpha_{tgt \to src} \leftarrow \text{Softmax}(a^T_{src \to tgt})$;
    $H^{s'} \leftarrow \text{Self-MHA}^s((1 - \gamma_2)\mathbf{Emb}(\mathbf{X}) + \gamma_2 \alpha_{tgt \to src} \mathbf{Emb}(\mathbf{Y}) + \mathbf{Pos}(\mathbf{X}))$;
    $H^{t'} \leftarrow \text{Self-MHA}^t(\text{IncMask}(\mathbf{Emb}(\mathbf{Y}) + \mathbf{Pos}(\mathbf{Y})))$;
    $a'_{src \to tgt} \leftarrow \text{Cross-MHA}(H^{t'}, H^{s'})$;
    $\mathcal{L}_{\text{EAL}} \leftarrow \text{Softmax}(\text{Linear}(H^{t'} + \alpha'_{src \to tgt} H^{s'}))$;
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{E2E}} + \mathcal{L}_{\text{EAL}}$;

tion mechanism ($\mathcal{K}$ heads), we average the alignment matrix of the $\mathcal{K}$ heads in each layer to obtain a final alignment matrix for each layer. For the alignment matrix of multiple layers, we experimentally found that choosing a specific layer L_A, instead of averaging the alignment matrices of all layers, can achieve better results.

4 Experiment

4.1 Setup

Datasets Our proposed method was evaluated on two typical translation tasks: rich-resource (WMT14 English-to-German (En→De), English-to-French (En→Fr), and WMT17 Chinese-to-English (Zh→En)) and low-resource (IWSLT14 German-to-English (De→En) and IWSLT13 French-to-English (Fr→En)). Please refer to Appendix A.1 for more dataset details.

Model Following [Vaswani *et al.*, 2017], we used the same **Transformer base/big** setting for rich-resource datasets. The Transformer.base model consists of a six-layer encoder and six-layer decoder. The number of heads, dimension of word embeddings, number of hidden states, and number of position-wise feedforward networks were 8, 512, 512, and 2048, respectively. The dropout was 0.1 and attention head was 8. In the Transformer.big model, the number of heads, dimension of word embeddings, number of hidden states, and number of position-wise feedforward networks were increased to 16, 1024, 1024, and 4096, respectively. For low-resource datasets, we used the **Transformer small** setting, which has a six-layer encoder and six-layer decoder, but the dimension of word embeddings, number of hidden states, and number of position-wise feedforward networks were 512, 512, and 1024, respectively. The dropout was 0.3 and attention head was 4. Word embeddings between the source, target, and output softmax embeddings were tied because it is a normal setting. For all experiments, the hyperparameters were optimized on a development set and then tested using only a single hyperparameter. We used a beam size of 4 and a length penalty of 0.6 for inference, and used *multi-bleu* to evaluate the quality of translation. In our EAL approach, the decoder layer for extracting the alignment weight was set to $L_A = 4$ and the number of warmup steps was set to $w = 20K$. We set the sampling ratio in our experiments to 20%, which means that in the self-training of EAL, 20% of the source token embeddings are replaced by embeddings using their aligned target embeddings, while the remaining 80% we keep the original inputs unchanged. γ_2 is set to 0.1, which is fixed during training. Our EAL method is only used during training to improve embedding alignment, and the decoding is same with the baseline.

Model	En→De		En→Fr		Zh→En	
	BLEU	Δ	BLEU	Δ	BLEU	Δ
Transformer.base	27.35	–	39.10	–	24.14	–
+EAL	**28.36**	1.01 ↑	**40.22**	1.12 ↑	**24.95**	0.81 ↑
JointAlign	28.80	–	41.95*	–	25.12*	–
Transformer.big	28.45	–	41.07	–	24.55	–
+EAL	**29.27**	0.82 ↑	**42.34**	1.27 ↑	**25.45**	0.90 ↑

Table 1: WMT En→De, En→Fr, and Zh→En BLEU scores on rich-resource datasets. * indicates that the results from our own reproduced model are reported.

Training All our models were trained on eight NVIDIA V100 GPUs. The learning rate setting strategy, which was the same as [Vaswani *et al.*, 2017], was $lr = d^{-0.5} \cdot min(step^{-0.5}, step \cdot warmup_{step}^{-1.5})$, where d is the dimension of embeddings, $step$ is the number of training steps, and $warmup_{step}$ is the number of warmup steps. In each training step, there was a set of sentence pairs containing approximately $4,096 \times 8$ source tokens and $4,096 \times 8$ target tokens. The value of label smoothing was set to 0.1. The learning rate was varied under a warmup strategy, with warmup steps of 8,000. In our experiments, the baseline is trained with as many steps as our model, that is, the total number of update steps is consistent.

4.2 Results

Table 1 presents the evaluation results of WMT14 En-De, En-Fr, and WMT17 Zh-En in a rich-resource setting. On Transformer.base, our EAL method achieved a stable improvement on all three datasets. This result shows that the self-training method can effectively learn alignment information that is useful for translation. On Transformer.big, the results of the EAL method show the same conclusion as on Transformer.base: EAL improved translation performance. Compared with JointAlign [Garg *et al.*, 2019b], our EAL method achieved better results without the help of an external alignment model, showing the effectiveness and convenience of our proposed method. In addition, the stable improvement achieved on the rich-resource datasets shows that the implicit alignment learning performed by the current baseline methods is not sufficient. Integrating EAL into training can effectively improve the performance of translation.

To explore the generalization of our proposed method, we also conducted experiments on the low-resource translation

Model	De→En		Fr→En	
	BLEU	Δ	BLEU	Δ
Transformer.small	34.49	–	42.95	–
+EAL	**35.82**	1.33 ↑	**44.41**	1.46 ↑

Table 2: BLEU scores on IWSLT14 De→En and IWSLT13 Fr→En datasets.

baseline; the results are reported in Table 2. As the table shows, the EAL method achieved improvements of 1.33 and 1.46. The comparison shows that (1) the EAL method achieved better results than the baseline on all the datasets without extra model parameters, and (2) the EAL method achieved the best results without the external alignment model. In particular, we find that the EAL method worked best on relatively small-scale datasets. Because small-scale datasets contain less bilingual information than large-scale datasets and easily cause overfitting problems, these results clearly demonstrate the effectiveness of our approach. Please refer to Appendix A.3 for case study and more detail analysis.

4.3 Alignment Evaluation

Although our work does not directly improve the alignment effect in cross-attention, we found that improving the source–target embedding alignment leads to an improvement in the cross-attention alignment indirectly, and the self-training process in our EAL also improves the cross-attention. We start from this cross-attention alignment to study the overall alignment improvement achieved by our model. We followed the practice of [Garg *et al.*, 2019b] and slightly modified the preprocessing pipeline for the WMT14 English-to-German task, to enable the evaluation of the alignment quality against the gold alignments provided by [Vilar *et al.*, 2006]. The alignment quality was evaluated by using the alignment error rate (AER), which was introduced in [Och and Ney, 2003].

Model	Precision	Recall	AER	BLEU
GIZA (word-based)	85.8	68.2	24.0	-
GIZA (BPE-based)	86.8	71.9	21.3	-
JointAlign [Garg *et al.*, 2019b]	66.6	69.1	32.2	28.50
+full-context	75.0	78.0	23.5	28.50
++GIZA supervised	78.8	81.7	19.8	28.80
Transformer.big (Layer Average)	32.7	33.9	66.7	28.45
Transformer.big (4th Layer)	32.9	34.2	66.5	28.45
+EAL	67.2	68.9	32.0	29.27

Table 3: WMT14 En→De results on the align and translate task. AER is used to evaluate alignment quality and BLEU to evaluate translation quality.

The cross-attention alignment evaluation results are shown in Table 3. First, we compared the alignment quality of averaging the cross-attention weights of all decoder layers and employing the L_A layer, as in EAL. We found that the AER of the $L_A = 4$ layer was less than that of averaging all layers; this shows that the top layer in the decoder may no longer contain word-level alignment information. The cross-attention in the EAL method is provided internally and it has also been optimized jointly, so the AER was greatly reduced. Compared with JointAlign [Garg *et al.*, 2019a], although our method had a larger AER, it still achieved a better BLEU score. This shows that the alignment in cross-attention is not the only factor that determines the translation effect: embedding alignment is also important for good translation. Please refer to Appendix A.2 for embedding alignment analysis.

Layer	De→En	Fr→En
1 (bottom)	35.03	43.85
2	35.35	44.12
3	35.57	44.30
4	**35.82**	**44.41**
5	35.31	43.81
6 (top)	34.97	43.46
Average	35.60	44.27

Table 4: BLEU scores of different alignment layers.

Pooling	De→En	Fr→En
Min	32.36	42.68
Max	35.68	44.15
Average	**35.82**	**44.41**

Table 5: BLEU scores of different pooling strategies with multi-head cross-attention.

4.4 Impact of L_A and Pooling Strategy on EAL

In proposed EAL approach, there are multiple alignment weights from the multiple stacked layers in the decoder. Therefore, to explore which layer is better for embedding alignment learning, we used IWSLT14 De-En and IWSLT13 Fr-En to conduct empirical exploration, the results of which are shown in Table 4. The results show that the model performance gradually increased from the bottom layer to the top layer, reached a maximum at $L_A = 4$, and then began to decline. This shows that the low layer is not the most suitable for cross-language alignment information because of the encoding of linguistic information, and the top layer is not suitable because it is close to the specific target.

Because cross-attention is implemented using multi-head attention, each head obtains an alignment weight. To obtain a single alignment weight, a pooling strategy is required. Common pooling strategies are max pooling, min pooling, and average pooling. The results of comparing these strategies are shown in Table 5. The comparison shows that max pooling and average pooling can improve performance, whereas min pooling has a negative effect. This is because the original intention of min pooling is inconsistent with alignment weight.

5 Conclusions

In this work, we have presented EAL methods for NMT, which randomly replace words or mixup with their aligned alternatives in another language when training. These methods are simple yet effective for helping NMT to learn alignment information from SMT. Results on both small-scale and large-scale datasets have verified the effectiveness of our methods. In the future, in addition to focusing on bilingual machine translation tasks, we are interested in extending our method to a multilingual scenario, which needs more complex replacement and training strategies. In addition, we plan to apply our approach to other cross-lingual NLP tasks.

A Appendix

A.1 Dataset Details

Rich-resource. We used the WMT14 En→De dataset with 4.5M sentence pairs for training. We randomly selected 4K of data from the training set as the validation set and used newstest2014 as the test set. The WMT14 En→Fr training set contained 36M bilingual sentence pairs, the newstest2012 and newstest2013 datasets were combined for validation, and newstest2014 was used as the test set. The WMT17 Zh→En training set contained 22M bilingual sentence pairs, and the newsdev2017 and newstest2017 datasets were used as the validation and test sets, respectively. The dataset was segmented by the BPE algorithm [Sennrich *et al.*, 2016] and the number of subword tokens in the shared vocabulary was 32K. Sentences longer than 300 subword tokens were removed from the training set.

Low-resource. The IWSLT14 De→En dataset contains 153K training sentence pairs. We randomly selected 7K of data from the training set as the validation set and used the combination of dev2010, dev2012, tst2010, tst2011, and tst2012 as the test set, containing 7K sentences, which were first preprocessed. The IWSLT13 Fr→En dataset contains about 200K training sentence pairs, and we used the tst2012 set for validation and the tst2013 set for testing. The BPE algorithm was used to process words into subwords, and the number of subword tokens in the shared vocabulary was 10K for both datasets.

A.2 Embedding Alignment Analysis

Because we assume that aligned word pairs appearing in the same position during training are helpful for forming bilingual embeddings that improve performance [Liu *et al.*, 2019], we investigated whether our approach is truly useful for bilingual embeddings. We randomly sampled some words and their corresponding aligned words to analyze the relations between them. Specifically, we calculated the cosine similarity between the embeddings of aligned words to determine the changes of bilingual embeddings. Formally, we have aligned word pairs $(\mathbf{x}_i, \mathbf{y}_j)$ and their embeddings $\mathbf{E}(\mathbf{x}_i) = (\mathbf{e}(\mathbf{x}_i)_1, \mathbf{e}(\mathbf{x}_i)_2, \cdots, \mathbf{e}(\mathbf{x}_i)_d)$ and $\mathbf{E}(\mathbf{y}_j) = (\mathbf{e}(\mathbf{y}_j)_1, \mathbf{e}(\mathbf{y}_j)_2, \cdots, \mathbf{e}(\mathbf{y}_j)_d)$, where d is the embedding dimension. Cosine similarity is defined as:

$$\cos\theta_{(\mathbf{E}(\mathbf{x}_i),\mathbf{E}(\mathbf{y}_j))} = \frac{\sum_{k=1}^{d} \mathbf{e}(\mathbf{x}_i)_k \cdot \mathbf{e}(\mathbf{y}_j)_k}{\sqrt{\sum_{k=1}^{d} \mathbf{e}(\mathbf{x}_i)_k^2} \cdot \sqrt{\sum_{k=1}^{d} \mathbf{e}(\mathbf{y}_j)_k^2}} \tag{9}$$

where $\theta_{(\mathbf{E}(\mathbf{x}_i),\mathbf{E}(\mathbf{y}_j))}$ is the angle between embedding pairs. We finally normalize the results to (0, 1): a larger value corresponds to a greater similarity between the two embeddings.

The results in Fig. 2 lead to the following observations. (1) The embedding vectors between aligned word pairs have a very strong positive correlation because the normalized cosine similarity values are all greater than 0.5. (2) The EAL methods significantly improve the positive correlation between aligned word pairs; this proves our hypothesis that switching aligned words is helpful for forming bilingual embeddings. (3) The EAL method improves the quality of bilingual embeddings by the largest ratio; this is consistent with the improvement in the translation BLEU score. (4) There may be some differences between the alignment found by the model and the alignment that people understand. The internal model tends to find the alignment that can maximize the BLEU score, which is not necessarily the real semantic alignment.

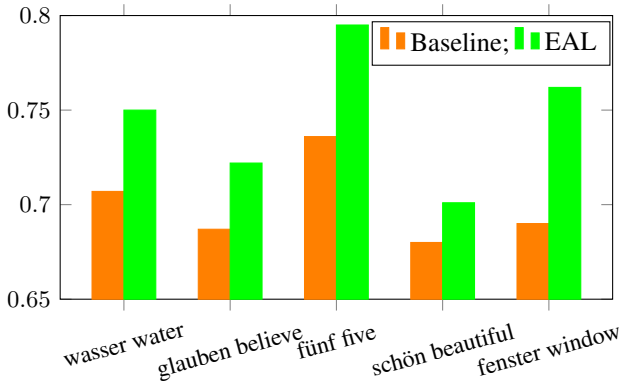

Figure 2: Cosine similarity between some bilingual embedding pairs in different models, trained on IWSLT14 De-En. The results have been normalized to (0, 1).

Src	*Wenn sie mir nur einen wunsch für die nächsten 50 jahre gestatten.*
Ref	*If you gave me only one wish for the next 50 years.*
Base	*If you only let me give you a wish for the next 50 years.*
Ours	*If you just allowed me to have a wish for the next 50 years.*
Src	*Verwischen wir die grenzen zwischen den verschiedenen bildern und lassen sie wie ein einzelnes foto aussehen, obwohl ein bild grundsätzlich hunderte von ebenen enthalten kann.*
Ref	*We erase the borders between the different images and make it look like one single image, despite the fact that one image can contain hundreds of layers basically.*
Base	*We expose the boundaries between the different images and let them look like a single photo, even though an image can basically contain hundreds of levels.*
Ours	*We wipe the boundaries between the different images and make them look like a single photograph, even though a picture can basically contain hundreds of levels.*

Table 6: Translation examples by the baseline and our proposed model on the IWSLT14 De→En dataset.

A.3 Case Study

In Table 6, we present two translation examples, by the baseline and by our best model, from the IWSLT14 German-to-English dataset. In the first example, active and passive relationships are reversed because of misalignment in the baseline model, whereas our proposed model corrects this mistake. In the second example, the verb *expose* is incorrectly used in the baseline model because of misalignment, but our proposed model uses the right verb—*wipe* and *erase* are synonyms. As the above examples demonstrate, our model does learn some alignment information.

References

[Al-Onaizan and Papineni, 2006] Yaser Al-Onaizan and Kishore Papineni. Distortion models for statistical machine translation. In *COLING*, 2006.

[Alkhouli and Ney, 2017] Tamer Alkhouli and Hermann Ney. Biasing attention-based recurrent neural networks using external alignment information. In *WMT*, 2017.

[Alkhouli *et al.*, 2018] Tamer Alkhouli, Gabriel Bretschner, et al. On the alignment problem in multi-head attention-based neural machine translation. In *WMT*, 2018.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, et al. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2014.

[Brown *et al.*, 1993] Peter F. Brown, Stephen Della Pietra, et al. The mathematics of statistical machine translation: Parameter estimation. *CL*, 1993.

[Chatterjee *et al.*, 2017] Rajen Chatterjee, Matteo Negri, et al. Guiding neural machine translation decoding with external knowledge. In *WMT*, 2017.

[Duan *et al.*, 2021] Chaoqun Duan, Kehai Chen, and othrs. Modeling future cost for neural machine translation. *TASLP*, 2021.

[Garg *et al.*, 2019a] Sarthak Garg, Stephan Peitz, et al. Jointly learning to align and translate with transformer models. In *EMNLP*, 2019.

[Garg *et al.*, 2019b] Sarthak Garg, Stephan Peitz, et al. Jointly learning to align and translate with transformer models. In *EMNLP*, 2019.

[Gehring *et al.*, 2017a] Jonas Gehring, Michael Auli, et al. A convolutional encoder model for neural machine translation. In *ACL*, 2017.

[Gehring *et al.*, 2017b] Jonas Gehring, Michael Auli, et al. Convolutional sequence to sequence learning. In *ICML*, 2017.

[He *et al.*, 2016] Wei He, Zhongjun He, et al. Improved neural machine translation with smt features. In *AAAI*, 2016.

[Koehn *et al.*, 2003a] Philipp Koehn, Franz J. Och, et al. Statistical phrase-based translation. In *NAACL*, 2003.

[Koehn *et al.*, 2003b] Philipp Koehn, Franz Josef Och, et al. Statistical phrase-based translation. In *NAACL*, 2003.

[Liu *et al.*, 2016] Lemao Liu, Masao Utiyama, et al. Neural machine translation with supervised attention. In *COLING*, 2016.

[Liu *et al.*, 2019] Xuebo Liu, Derek F Wong, et al. Shared-private bilingual word embeddings for neural machine translation. In *ACL*, 2019.

[Luong *et al.*, 2015] Minh-Thang Luong, Hieu Pham, et al. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[Mi *et al.*, 2016] Haitao Mi, Zhiguo Wang, et al. Supervised attentions for neural machine translation. In *EMNLP*, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, et al. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[Och and Ney, 2003] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *CL*, 2003.

[Och *et al.*, 2004] Franz Josef Och, Daniel Gildea, et al. A smorgasbord of features for statistical machine translation. In *NAACL*, 2004.

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, et al. Neural machine translation of rare words with subword units. In *NAACL*, 2016.

[Shaw *et al.*, 2018] Peter Shaw, Jakob Uszkoreit, et al. Self-attention with relative position representations. In *NAACL*, 2018.

[Song *et al.*, 2020] Kai Song, Kun Wang, et al. Alignment-enhanced transformer for constraining NMT with pre-specified translations. In *AAAI*, 2020.

[Stahlberg *et al.*, 2016] Felix Stahlberg, Eva Hasler, et al. Syntactically guided neural machine translation. In *ACL*, 2016.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, et al. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[Tillmann, 2004] Christoph Tillmann. A unigram orientation model for statistical machine translation. In *NAACL*, 2004.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, et al. Attention is all you need. In *NIPS*, 2017.

[Vilar *et al.*, 2006] David Vilar, Maja Popović, et al. Aer: Do we need to "improve" our alignments? In *IWSLT*, 2006.

[Wang *et al.*, 2017] Xing Wang, Zhengdong Lu, et al. Neural machine translation advised by statistical machine translation. In *AAAI*, 2017.

[Wang *et al.*, 2018] Xinyi Wang, Hieu Pham, et al. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *EMNLP*, 2018.

[Zhang *et al.*, 2017] Jinchao Zhang, Mingxuan Wang, et al. Incorporating word reordering knowledge into attention-based neural machine translation. In *ACL*, 2017.

[Zhang *et al.*, 2021] Jiacheng Zhang, Huanbo Luan, et al. Neural machine translation with explicit phrase alignment. *TASLP*, 2021.

[Zhao *et al.*, 2018] Yang Zhao, Yining Wang, et al. Phrase table as recommendation memory for neural machine translation. In *IJCAI*, 2018.

[Zheng *et al.*, 2019] Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. Dynamic past and future for neural machine translation. In *EMNLP*, 2019.

[Zhou *et al.*, 2017] Long Zhou, Wenpeng Hu, et al. Neural system combination for machine translation. In *ACL*, 2017.