EFFICIENT NEURAL MACHINE TRANSLATION WITH PRIOR WORD ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior word alignment has been shown indeed helpful for better translation, if such prior is good enough and can be acquired in a convenient way at the same time. Traditionally, word alignment can be learned through statistical machine translation (SMT) models. In this paper, we propose a novel method that infuses prior word alignment information into neural machine translation (NMT) to provide hints or guidelines for the target sentence at running time. To this end, previous works of similar approaches should build dictionaries for specific domains, or constraint the decoding process, or both. While being effective to some extent, these methods may greatly affect decoding speed and hurt translation flexibility and efficiency. Instead, this paper introduces an enhancement learning model, which can learn how to directly replace specific source words with their target counterparts according to prior alignment information. The proposed model is then inserted into a neural MT model and augments MT input with the additional target information from the learning model in an effective and more efficient way. Our novel method achieves BLEU improvements (up to 1.1) over a strong baseline model on English-Korean and English-Romanian translation tasks.

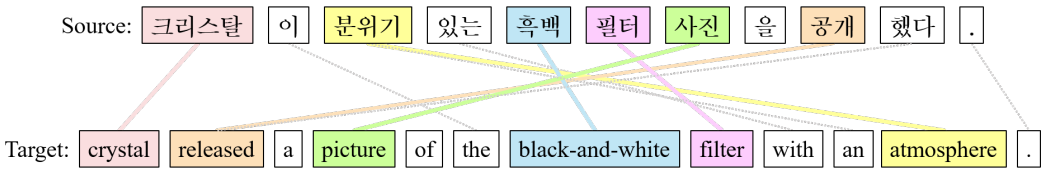
1 INTRODUCTION

As neural machine translation (NMT) models have become the dominant approach in the machine translation task, explicit word alignment, which is an essential intermediary result from the training of most statistical machine translation (SMT) models, seems becoming increasingly obsolete. While SMT can control the output using the word alignment information, it is difficult for the less interpretable NMT systems to apply it effectively because the latter takes an end-to-end learning way. Although unnecessary for training NMT models, word alignment is still useful for translation error analysis (Ding et al., 2017), understanding black-box NMT systems (Li et al., 2019; 2020a), providing guidance for NMT decoding directly (Alkhouli et al., 2018; Song et al., 2020), and extracting better alignment from NMT models (Garg et al., 2019; Chen et al., 2020; Zenkel et al., 2019; 2020).

With the guidance of good enough known word alignment, replacing some words in the source sentence with semantically corresponding words in the target language leads to better translation or user-desired translation, and it is also known as a tip to use the translator well. As in Figure 1, we can see that an open translation system [1] generates a better translation closer to the target sentence when some words of the target sentence are provided in the source sentence. In order words, a user can use specific alignment, such as 공개 ↔ *released* and 사진 ↔ *picture*, to get a desired translation. The case in Figure 1 happens because word alignment between source and target sentences more or less holds no matter how the model acquires such alignment. Besides, not all word alignment may help and only those good enough word alignment can truly enhance the model. When the concerned language pair shares a large vocabulary, such good enough alignment may be easily obtained and then conveniently works in the proposed early substituting way. This work will right explore an effective way to figure out those 'good' enough certain alignment for neural machine translation enhancement.

Previous studies of similar approaches can be largely divided into two categories: *dictionary-guided decoding* and *augmenting MT input with its corresponding target information*. The former is to

[1] https://translate.google.com



Systems	Sentences
Source	크리스탈이 분위기 있는 흑백 필터 사진을 공개했다.
►Translator Out	crystal unveiled* an atmospheric black and white filter photo*.
Source with Target	크리스탈이 분위기 있는 흑백 필터 picture을 released했다.
►Translator Out	crystal released a picture of atmospheric black and white filter.

Figure 1: A Korean-to-English translation example of inputting a source sentence, that replaces the input word with its corresponding target one, in an open translation system. The colorful line between the source and target at the top represents 'good' alignment, and the gray dotted line is a trivial one. The underline denotes the target word corresponding to the source one, and the asterisk* is the word to correct using the 'good' alignment.

leverage pre-specified translations to guide the decoding procedure in the modified NMT that has additional attention head to learn the word alignment based on supervision from external word alignment signals (Alkhouli et al., 2018; Song et al., 2020). The latter is to augment MT input with pre-defined dictionary entries for a specific domain and let NMT models learn, at training time, how to use the corresponding target term when provided in the source sentence (Dinu et al., 2019; Park & Zhao, 2019).

In this paper, we focus on integrating word alignment information into NMT systems in an effective and more efficient way, without building special dictionary entries beforehand. To this end, we exploit the alignment learned by SMT model, insert an Alignment-Based Word Substitution (**ABWS**) model into NMT system, and the goal of the model is to learn how to replace the input word with the target one at training time. Note that the input of the ABWS model is the original source sentence and the reference is the modified source sentence with some of them replaced by the target word according to the word alignment. At inference time, the final hidden state of the trained ABWS model is integrated into NMT models to provide additional target information for MT input. In our experiment, we use Transformer (Vaswani et al., 2017) as the baseline model that relies on self-attention mechanisms and has achieved state-of-the-art performance in recent NMT tasks.

2 Related Work

Word alignment is no longer an indispensable component in training NMT models, but recently there has been a resurgence of interest in the community to study word alignment for NMT models due to its better interpretability. For example, previous works used word alignment information to interpret the end-to-end NMT system and analyze translation errors or to extract better alignments from the learned NMT models. There have been also several studies that leverage word alignment to guide NMT decoding directly, and especially Alkhouli et al. (2018) described this approach as a new downstream task of leveraging word alignment (`dictionary-guided decoding task`). In this study, we focus on infusing SMT offered word alignment into NMT system to improve translation performance.

Alkhouli et al. (2018) added a special alignment head, which conditions a lexical model on the alignment information, to the multi-head source-to-target attention module of Transformer decoder. The use of a separate alignment model adds significant computational overhead to the decoding process, requiring special handling to optimize speed. And Song et al. (2020) proposed an approach that introduces a dedicated head in the multi-head Transformer architecture to capture external supervision signals. Besides, these two studies constraint the decoding process to correct translation errors with pre-defined dictionaries and achieve significant improvement on translation tasks. Dinu et al. (2019); Park & Zhao (2019) also used pre-defined dictionaries for a specific domain to let NMT model learn

how to use dictionary entries when provided with the input, but they do not constraint the decoding procedure and adopt a non-coercive method that augments MT input with additional information. Furthermore, Dinu et al. (2019) showed that constrained decoding hurt translation performance in their experiments. While our goal resembles that of Dinu et al. (2019); Park & Zhao (2019) (converting MT input to augment it with pre-defined dictionaries for a specific domain), our proposed method does not need to build the pre-defined dictionaries and not limit to a specific domain, all it needs is to feed bilingual word alignment during training.

After NMT has become the dominant MT approach, there have been a variety of studies that integrate external resources (i.e. lemmas, POS tags, named entity, and other linguistic features) into NMT systems. Sennrich & Haddow (2016) proposed a simple but novel method that augments input embedding with its corresponding linguistic features through concatenation operation. Li et al. (2020b) presented three ways to integrate the inputs' compressed sentence into NMT systems. Correspondingly, we propose three integration methods to infuse the target information generated from the ABWS model into NMT models.

3 MODEL

In this section, we first describe the Transformer (Vaswani et al., 2017) for machine translation. Then we propose our core model, Alignment-Based Word Substitution (**ABWS**) model that replaces part of the source sentence with the corresponding target one according to prior word alignment. Furthermore, we introduce three strategies to fuse ABWS output into NMT system: Source-side Addition (**SA**), Source-side Context Gate (**SCG**), and Target-side Context Gate (**TCG**). Figure 2 shows the architecture of our proposed model with TCG integration strategy.

3.1 TRANSFORMER

A Transformer architecture follows the encoder-decoder paradigm (Sutskever et al., 2014) and has N stacked encoder layers and decoder layers that rely entirely on self-attention networks. A sequence of input words is first fed into a word embedding layer to get word embeddings. Then positional information is injected into the embeddings. The word embeddings are fed into the encoder layer that consists of two sub-modules, namely a self-attention module and a feed-forward module. The self-attention module first creates a query matrix $\boldsymbol{Q}$, a key matrix $\boldsymbol{K}$, and a matrix-vector $\boldsymbol{V}$ from each of the word embeddings and product an output matrix as follows:

$$SelfAttn(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Softmax(\frac{\boldsymbol{QK}}{d_{model}})\boldsymbol{V}, \tag{1}$$

where d_{model} denotes the dimensions of the model. To get a better meaning and context, the Transformer uses different attention heads that are computed parallelly and independently. Multi-head attention is computed from the concatenation of n attention heads output $head_i$:

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, ..., head_i)\boldsymbol{W^O}, \tag{2}$$

$$head_i = SelfAttn(\boldsymbol{QW_i^Q}, \boldsymbol{KW_i^K}, \boldsymbol{VW_i^V}), \tag{3}$$

where $\boldsymbol{W^O} \in \mathbb{R}^{d\times d}$, $\boldsymbol{W_i^Q}, \boldsymbol{W_i^K}, \boldsymbol{W_i^V} \in \mathbb{R}^{d\times \frac{d}{n}}$ are parameter matrices. Another sub-module, a feed-forward (FFN) network, consists of two fully-connected layers with a `relu` activation in between. Each sub-module is connected with a residual connection and followed by a layer normalization. The final hidden representation $\boldsymbol{H}_{src}$ of a nth encoder layer is generated through above-mentioned two sub-modules as:

$$\boldsymbol{H}_{src}^n = FFN(MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})). \tag{4}$$

Each decoder layer consists of a self-attention module, an encoder-decoder attention module, and a feed-forward module. To preserve autoregressive property, the self-attention module of the decoder layer, modified to prevent positions to attend to subsequent positions, is known as Masked multi-head attention. And an encoder-decoder attention module matches the encoder's input to the decoder's input, allowing the decoder to decide which encoder input is relevant to put a focus on. The final hidden representation $\boldsymbol{H}_{final}$ of nth decoder layer is generated through above-mentioned

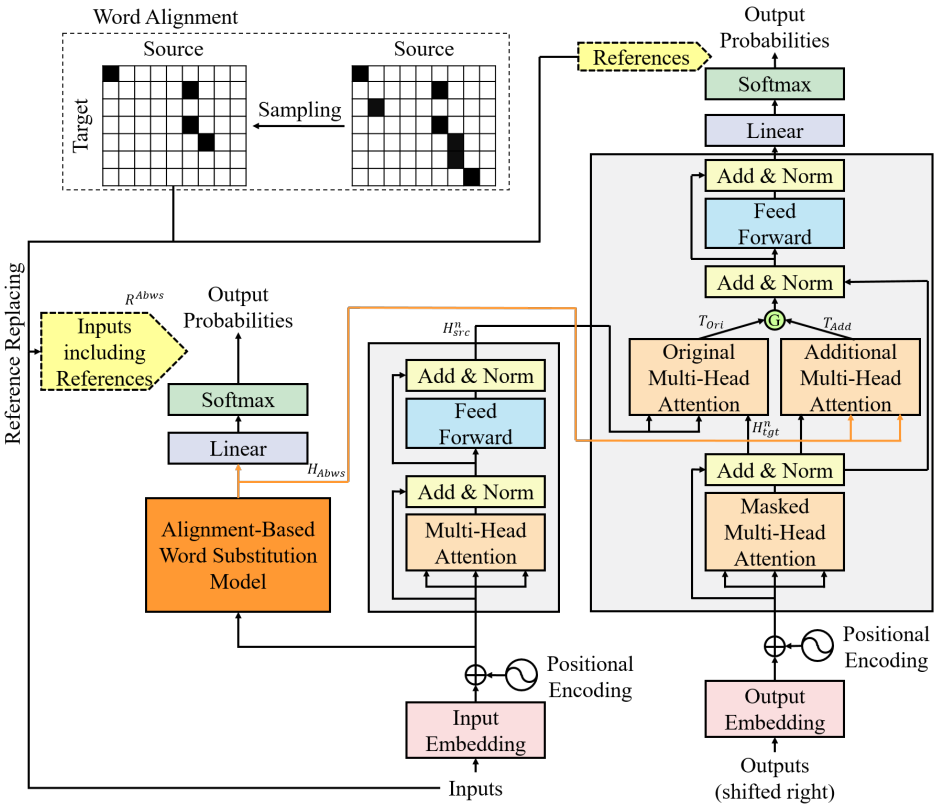


Figure 2: The architecture of our proposed model.

three sub-modules as:

$$\boldsymbol{H}^n_{final} = FFN(MultiHead(\boldsymbol{H}^n_{tgt}, \boldsymbol{H}^N_{src}, \boldsymbol{H}^N_{src})), \tag{5}$$

$$\boldsymbol{H}^n_{tgt} = MaskedMultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})). \tag{6}$$

Finally, a softmax layer based on the decoder's last layer produces a probability distribution over the target vocabulary:

$$p(\boldsymbol{y}_t|\boldsymbol{y}_1, ..., \boldsymbol{y}_{t-1}, \boldsymbol{x}) = Softmax(\boldsymbol{H}^N_{final}\boldsymbol{W}^F), \tag{7}$$

where $\boldsymbol{W}^F$ is the learned weight matrix, $\boldsymbol{x}$ is the source sentence, and $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_t$ is the target words.

3.2 ALIGNMENT-BASED WORD SUBSTITUTION MODEL

We propose Alignment-Based Word Substitution model that learns, at training time, how to align the source word with the target one while target words are provided in the source sentence. The model consists of 2 stacked base Transformer's encoder layers and is jointly trained with the NMT model, and the ABWS model input and reference are as follows:

- Input $\boldsymbol{I}_{L\times V}$: Original sentence identical to the input of the NMT model,
- Reference $\boldsymbol{R}^{Abws}_{L\times V}$: Modified sentence that generated by replacing the source word with the target one according to the prior alignment information,

where L is the length of the source sentence and V is the size of the shared vocabulary. At the inference time, the last hidden state of the trained model is incorporated into the input of the NMT model.

As our heuristics are applied to the use of alignment in the word substitution and model training, we distinguish three cases of bilingual word alignment. : one-to-null (unaligned word), one-to-one, and one-to-many (multi-aligned word). For one-to-one alignment, we simply replace the source word with the target one in the sentence. Since unaligned source words are not replaced, this model does not need special handling for them. However, there are some difficulties in processing a multi-aligned word. Unlike normal single-label classification models where class labels are mutually exclusive, the model should be able to classify multi-label data. In other words, if one source word is replaced with multiple target words, this problem cannot be approached with the single-label classification task like the typical sequence generation model. Furthermore, some of the target words may not be the key meaning of the corresponding source one. Therefore, we present three solutions to these issues: (1) Like the noising method of the existing pre-training language models (Devlin et al., 2019; Yang et al., 2019), the replacement process is performed with randomly sampled pairs at a certain ratio from all alignment pairs for each batch. The ratio p for sampling alignment pairs will be empirically determined. (2) It gives the model the flexibility to either learn core meaning from aligned target words or to deviate from fixed source-target word substitution pairs. The parameter of the output linear layer and the parameter of word embedding are shared, and only the parameters of the model are trained. At the inference time, the final hidden state of the model, not the output generated by the `argmax` operation, is directly integrated into the NMT model. This can preserve information about multi-aligned words. (3) Inspired by Garg et al. (2019), we implement multi-label classification for the multi-aligned words. Formally, let $\boldsymbol{O}^{Abws}_{L\times V}$ be the output of the model. We minimize the Kullback-Leibler divergence between $\boldsymbol{R}^{Abws}_{l,v}$ and $\boldsymbol{O}^{Abws}_{l,v}$ which is equivalent to optimizing the following cross-entropy loss $\mathcal{L}_a$:

$$\mathcal{L}_a(\boldsymbol{O}^{Abws}) = \frac{1}{V}\sum_{l=1}^{L}\sum_{v=1}^{V} d_{l,v}\boldsymbol{R}^{Abws}_{l,v}\log(\boldsymbol{O}^{Abws}_{l,v}), \tag{8}$$

where $d_{l,v}$ (duplicate alignment penalty) is the inverse frequency of each target index for each alignment sequence, and the goal of this penalty is to exclude words that are repeated too often but have little meaning (i.e. the, a, and so on) from model training. We train our model to minimize $\mathcal{L}_a$ in conjunction with the standard translation loss $\mathcal{L}_t$. The overall loss $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}_t + \lambda\mathcal{L}_a(\boldsymbol{O}^{Abws}), \tag{9}$$

where λ is a hyper-parameter, which is set to 0.1 in our experiments.

3.3 Integration strategies

While many different ways have been explored to augment MT input with additional information, we consider Three strategies to incorporate the last hidden state of the ABWS model into NMT system: SA, SCG, and TCG. Given the last hidden state of the ABWS model $\boldsymbol{H}_{Abws} \in \mathbb{R}^{l\times d}$ and input word embeddings $\boldsymbol{S} \in \mathbb{R}^{l\times d}$, where l is the length of source sentence , these three integration strategies are performed as follows.

Source-side Addition (SA) is to add $\boldsymbol{S}$ to $\boldsymbol{H}_{Abws}$:

$$Add(\boldsymbol{S}, \boldsymbol{H}_{Abws}) = \boldsymbol{S} + \boldsymbol{H}_{Abws}. \tag{10}$$

Source-side Context Gate (SCG) is to use a context gate $\boldsymbol{G} \in \mathbb{R}^{l\times d}$ for fusing $\boldsymbol{S}$ and $\boldsymbol{H}_{Abws}$:

$$ConGate(\boldsymbol{S}, \boldsymbol{H}_{Abws}) = \boldsymbol{G}\odot\boldsymbol{S} + (1. - \boldsymbol{G})\odot\boldsymbol{H}_{Abws}, \tag{11}$$

$$\boldsymbol{G} = \sigma(MLP([\boldsymbol{S}; \boldsymbol{H}_{Abws}])), \tag{12}$$

where σ is the logistic sigmoid function, $\odot$ is the element-wise multiplication, and $[;]$ denotes the concatenation operation.

Target-side Context Gate (TCG) is to introduce an additional encoder-decoder multi-head attention layer for integrating $\boldsymbol{H}_{Abws}$ into the decoder layer:

$$TCG(\boldsymbol{S}, \boldsymbol{H}_{Abws}) = ConGate(\boldsymbol{T}_{Add}, \boldsymbol{T}_{Ori}), \tag{13}$$

$$\boldsymbol{T}_{Add} = AddEncDecMultiHead(\boldsymbol{H}^{n}_{tgt}, \boldsymbol{H}^{N}_{Abws}, \boldsymbol{H}^{N}_{Abws})), \tag{14}$$

$$\boldsymbol{T}_{Ori} = OriEncDecMultiHead(\boldsymbol{H}^{n}_{tgt}, \boldsymbol{S}^{N}, \boldsymbol{S}^{N})), \tag{15}$$

$$\boldsymbol{H}^{N}_{Abws} = FFN(EncMultiHead(\boldsymbol{H}^{Abws}, \boldsymbol{H}^{Abws}, \boldsymbol{H}^{Abws})), \tag{16}$$

$$\boldsymbol{S}^{N} = \boldsymbol{H}^{N}_{src} = FFN(EncMultiHead(\boldsymbol{S}, \boldsymbol{S}, \boldsymbol{S})), \tag{17}$$

Table 1: Corpora statistics

Corpus	Train	Valid	Test	BPE
English↔Romanian	598K	1999	1000	50K
Korean↔English	767K	2000	2000	32K/32K

where $TCG(\boldsymbol{S}, \boldsymbol{H}_{Abws})$ is fed to the FFN module of the decoder layer, and $AddEnDeMultiHead$ is an additional encoder-decoder multi-head attention layer that is identical with the original one.

4 EXPERIMENTS

To verify that the proposed method is effective, we perform experiments on two language pairs: Korean-English (KO↔EN) and English-Romanian (EN↔RO). For all translation tasks, we use BLEU (Papineni et al., 2002) for the evaluation of translation quality. All the model training is on a single NVIDIA Tesla V100 GPU.

4.1 DATASETS

For EN↔RO translation, we use the WMT16 news datasets. For KO↔EN translation, we use news dataset that provided by AIHub [2] and split the dataset for the validation and test data. We tokenize the RO↔EN corpus using Moses (Koehn et al., 2007) and perform joint source and target Byte-Pair-Encoding (Sennrich et al., 2016) to a vocabulary of 50K tokens. For KO↔EN dataset, we use KoNLPy [3] toolkit for Korean sentence tokenization and apply 32k BPE merge operation for each language separately. Table 1 shows the statistics of datasets used in our experiments.

4.2 SETUP

For Bilingual word aligner, we use the MGIZA++ [4] (Gao & Vogel, 2008), a parallel implementation of GIZA++, and FastAlign [5] toolkit with default parameters. We align the bilingual training corpora with FastAlign for all language pairs. For RO↔EN translation task, we additionally use word alignments produced by MGIZA++ to compare two different word aligners. Both FastAlign and GIZA are used with default settings and all the training corpora are in subword format. Furthermore, the pairs of sentences with an error (i.e. zero sentences) are pruned in the process of generating alignment.

In all experiments for this task, we train all models using a `base` Transformer configuration with an embedding size of 512, 6 encoder and decoder layers, 8 attention heads, shared source and target embeddings, the standard `relu` activation function and sinusoidal positional embedding. We train with a batch size of 3500 tokens and use the validation translation loss for early stopping and update parameters every 8 batches. Furthermore, We optimize the model parameters using Adam optimizer with a learning rate 7e-4 β_1= 0.9 and $\beta_2 = 0.98$, learning rate warm-up over the first 4000 steps. Additionally, we apply label smoothing with a factor of 0.1. We average over the last 5 checkpoints and run inference with a beam size of 5. All our experiments were performed using the Torch-based toolkit, Fairseq(-py) (Ott et al., 2019).

4.3 EXPERIMENTAL RESULTS

Table 2 shows the BLEU evaluation of our systems. For the experimental results, we made the following observation: (1) For the integration strategy, all SA, SCG, and TCG outperformed the baseline Transformer. Among the three integration strategies, TCG had the best performance in most translation tasks. It means that integration at the target-side is better than at the source-side.

[2] http://www.aihub.or.kr/

[3] https://konlpy.org/en/latest/

[4] https://github.com/moses-smt/mgiza

[5] https://github.com/clab/fast align

Table 2: Experimental results on RO↔EN (WMT16) and KO↔EN (AIHub) translation tasks.

System	RO→EN		EN→RO		EN→KO	KO→EN
Transformer	35.06		27.01		35.63	39.50
	FastAlign	GIZA++	FastAlign	GIZA++	FastAlign	FastAlign
SA	35.33	35.30	27.18	27.44	36.01	40.37
SCG	35.40	**35.65**	27.36	27.51	36.11	40.46
TCG	**35.43**	35.50	**27.53**	**27.59**	**36.19**	**40.63**

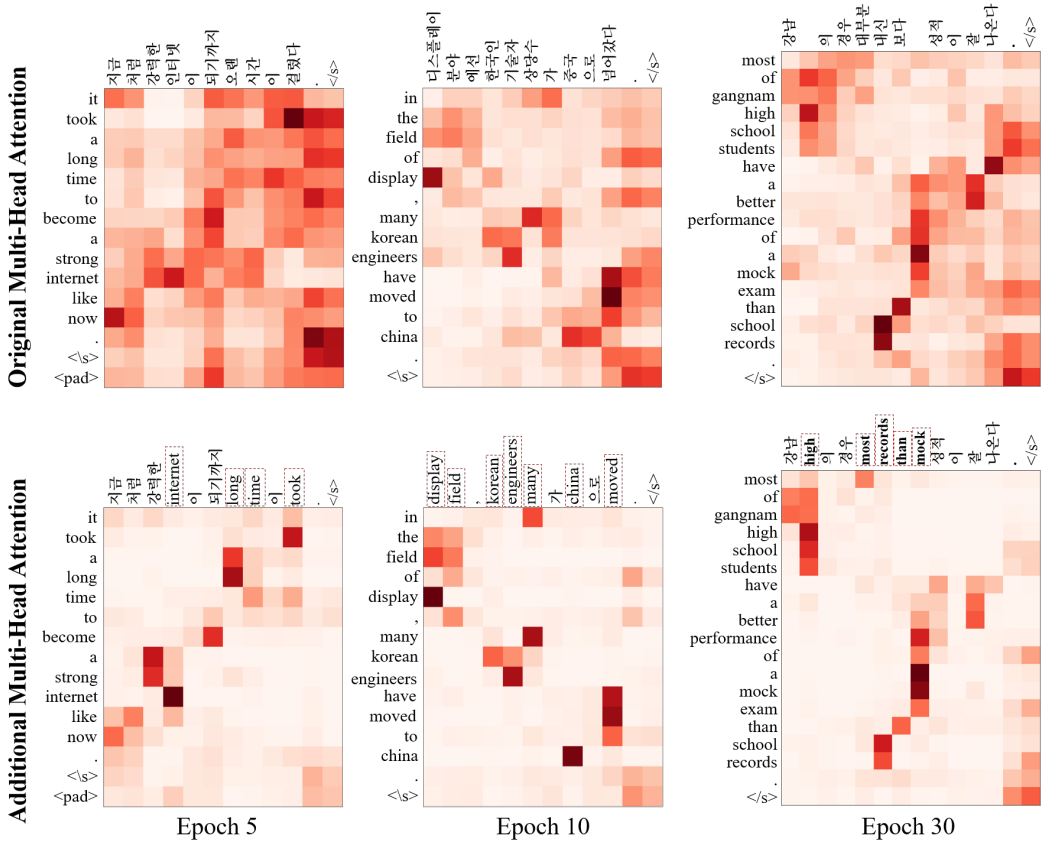


Figure 3: Comparison of attention probabilities extracted from the original encoder-decoder multi-head attention and the additional one of our proposed model with TCG integration on KO→EN translation task. The horizontal axis and the vertical axis represent the source sentence and the target sentence, respectively. The red box denotes the substituted word in the source sentence.

In addition, the parameters of our proposed model with the three integration strategies increased 9 (SA) 20 (TCG)% over the baseline Transformer. (2) For the auto-aligner, the results showed that word alignment information of GIZA++ yields a better performance improvement than FastAlign with a slight difference on EN↔RO translation tasks. This indicates that better alignment information can lead to better alignment information in our model. (3) Among translation tasks, the performance gain of the KO→EN task is noticeably the largest.

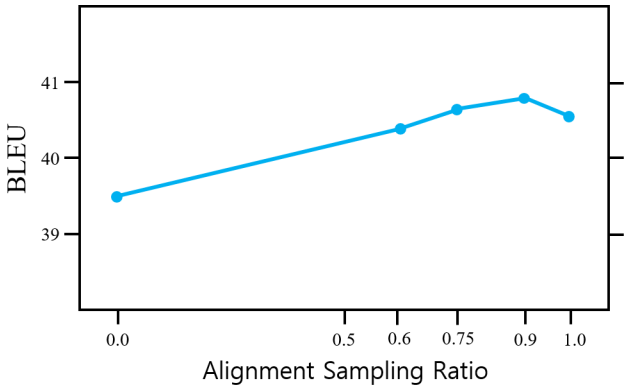


Figure 4: Performances on KO→EN translation task with different alignment sampling ratios

5 ANALYSIS

5.1 EVALUATING ALIGNMENT SAMPLING RATIO p

In order to verify the impact of different ratios p for sampling alignment pairs on translation quality, we conducted experiments on KO→EN translation task with NMT model using TCG integration strategy. When the sampling ratio p = 0.0, it means no use alignment information. And when the compression ratio p = 1.0, it is equivalent to leveraging all word alignment information. The experimental results are shown in Figure 4 shows the experimental results. Hence, the alignment sampling ratio is set to 0.9 for the best results in our experiments.

5.2 EVALUATING REPLACEMENT PROCESS

In this subsection, We evaluated the effectiveness of the replacement process with the NMT model using TCG integration strategy on the KO→EN translation task. To evaluate how our replacement process helps translation, we compared alignments obtained by layer-wise averaging of attention probabilities. Garg et al. (2019) showed that the attention probabilities from the penultimate layer naturally tend to learn alignments over all layers on three translation tasks. Hence, we used the attention probabilities from the penultimate layer in this experiment. Figure 3 shows the attention probabilities of two different multi-head attention during model training. In particular, the horizontal axis of the additional multi-head attention is the modified sentence generated by applying the `argmax` operation to the output probabilities of the ABWS model. Note that for better interpretation, unlike model training, we used output probabilities, not the final hidden state of the ABWS model. We can see that the additional multi-head attention module, directly connected to the ABWS model, catches relatively accurate alignment even at the beginning of training. This is especially true when the substituted word is included in the target sentence. It indicates that our method is guiding the NMT model where to attend in a smooth but effective way. Furthermore, many previous works noted that the word alignment information derived from Transformer is quite erroneous (Koehn & Knowles, 2017; Li et al., 2019; Ding et al., 2019). Although not directly absorbing alignment signals into NMT model (Garg et al., 2019; Zenkel et al., 2019; 2020), our proposed model can still effectively acquire clear alignment information.

6 CONCLUSION

This work presents a novel solution for the effective and efficient fusion of NMT and SMT. Especially, we explore an efficient way of exploiting prior word alignment offered by SMT models for NMT enhancement during the model training phase instead of constraint decoding in previous work which may slow down the inference. In detail, to augment NMT input, we design an extra model that learns how to replace specific source words with their corresponding target ones according to the prior word alignment. Our method helps yield significant performance improvement compared to a strong baseline on two translation tasks, which verifies the effectiveness of the proposed method.

REFERENCES

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 177–185, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6318. URL https://www.aclweb.org/anthology/W18-6318.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. Accurate word alignment induction from neural machine translation. 2020. URL https://arxiv.org/abs/2004.14837.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 1–12, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5201. URL https://www.aclweb.org/anthology/W19-5201.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume Volume 1: Long Papers, pp. 1150–1159. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1106. URL https://www.aclweb.org/anthology/P17-1106.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3063–3068, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1294. URL https://www.aclweb.org/anthology/P19-1294.

Qin Gao and Stephan Vogel. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pp. 49–57, Columbus, Ohio, 2008. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W08-0509.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly learning to align and translate with transformer models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, November 2019. URL https://arxiv.org/abs/1909.02074.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, Vancouver, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://www.aclweb.org/anthology/W17-3204.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic, 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

Jierui Li, Lemao Liu, Huayang Li, Guanlin Li, Guoping Huang, and Shuming Shi. Evaluating explanation methods for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020a.

Xintong Li, Guanlin Li, Max Liu, Lemao nad Meng, and Shuming Shi. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1293–1303. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1124. URL https://www.aclweb.org/anthology/P19-1124.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. Explicit sentence compression for neural machine translation. In *Proceedings of AAAI-2020*, 2020b. URL http://arxiv.org/abs/1912.11980.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Jeonghyeok Park and Hai Zhao. Korean-to-chinese machine translation using chinese character as pivot clue. In *Proceedings of 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)*, pp. 558–566, Hakodate, Japan., 2019.

Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pp. 83–91, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2209. URL https://www.aclweb.org/anthology/W16-2209.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pp. 1715–1725, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, and Min Zhang. Alignment-enhanced transformer for constraining nmt with pre-specified translations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. doi: https://doi.org/10.1609/aaai.v34i05.6418.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 6000—6010, 2017. URL http://arxiv.org/abs/1706.03762.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32*, pp. 5753–5763. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

Thomas Zenkel, Joern Wuebker, and John Denero. Adding interpretable attention to neural translation models improves word alignment. volume abs/1901.11359, 2019. URL http://arxiv.org/abs/1901.11359.

Thomas Zenkel, Joern Wuebker, and John Denero. End-to-end neural word alignment outperforms giza++. 2020. URL https://arxiv.org/abs/2004.14675.