MASK-ALIGN: Self-Supervised Neural Word Alignment

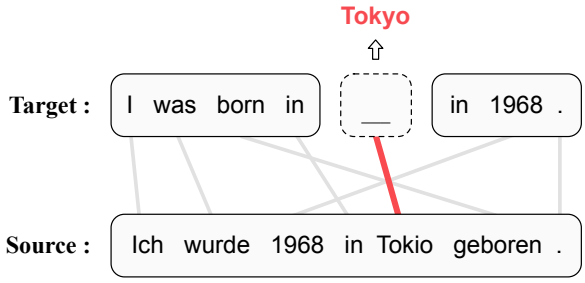
Chi Chen[1,3,4], **Maosong Sun**[1,3,4,5], **Yang Liu**[*1,2,3,4,5]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[4]Beijing National Research Center for Information Science and Technology
[5]Beijing Academy of Artificial Intelligence

Abstract

Word alignment, which aims to align translationally equivalent words between source and target sentences, plays an important role in many natural language processing tasks. Current unsupervised neural alignment methods focus on inducing alignments from neural machine translation models, which does not leverage the full context in the target sequence. In this paper, we propose MASK-ALIGN, a self-supervised word alignment model that takes advantage of the full context on the target side. Our model parallelly masks out each target token and predicts it conditioned on both source and the remaining target tokens. This two-step process is based on the assumption that the source token contributing most to recovering the masked target token should be aligned. We also introduce an attention variant called *leaky attention*, which alleviates the problem of high cross-attention weights on specific tokens such as periods. Experiments on four language pairs show that our model outperforms previous unsupervised neural aligners and obtains new state-of-the-art results.[1]

1 Introduction

Word alignment is an important task of finding the correspondence between words in a sentence pair (Brown et al., 1993) and used to be a key component of statistical machine translation (SMT) (Koehn et al., 2003; Dyer et al., 2013). Although word alignment is no longer explicitly modeled in neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017), it is often leveraged to analyze NMT models (Tu et al., 2016; Ding et al., 2017). Word alignment is also used in many other scenarios such as imposing lexical constraints on the decoding process (Arthur et al., 2016; Hasler et al., 2018), improving automatic post-editing (Pal et al., 2017) , and providing guidance for translators in computer-aided translation (Dagan et al., 1993).



Induced alignment link: **Tokio - Tokyo**

Figure 1: An example of inducing an alignment link for target token "Tokyo" in MASK-ALIGN. First, we mask out "Tokyo" and predict it with source and other target tokens. Then, the source token "Tokio" that contributes most to recovering the masked word (highlighted in red) is chosen to be aligned to "Tokyo".

Compared with statistical methods, neural methods can learn representations end-to-end from raw data and have been successfully applied to supervised word alignment (Yang et al., 2013; Tamura et al., 2014). For unsupervised word alignment, however, previous neural methods fail to significantly exceed their statistical counterparts such as FAST-ALIGN (Dyer et al., 2013) and GIZA++ (Och and Ney, 2003). Recently, there is a surge of interest in NMT-based alignment methods which take alignments as a by-product of NMT systems (Li et al., 2019; Garg et al., 2019; Zenkel et al., 2019, 2020; Chen et al., 2020). Using attention weights or feature importance measures to induce alignments for to-be-predicted target tokens, these methods outperform unsupervised statistical aligners like GIZA++ on a variety of language pairs.

Although NMT-based unsupervised aligners have proven to be effective, they suffer from two major limitations. First, due to the autoregressive

*Corresponding author
[1]Code can be found at https://github.com/THUNLP-MT/Mask-Align.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4781–4791
August 1–6, 2021.

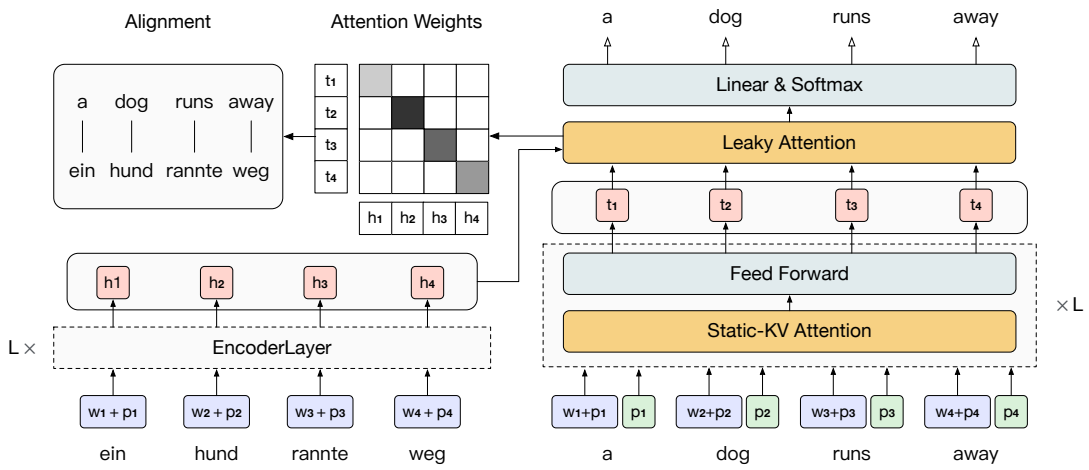


Figure 2: The architecture of MASK-ALIGN.

property of NMT systems (Sutskever et al., 2014), they only leverage part of the target context. This inevitably brings noisy alignments when the prediction is ambiguous. Consider the target sentence in Figure 1. When predicting "Tokyo", an NMT system may generate "1968" because future context is not observed, leading to a wrong alignment link ("1968", "Tokyo"). Second, they have to incorporate an additional guided alignment loss (Chen et al., 2016) to outperform GIZA++. This loss requires pseudo alignments of the full training data to guide the training of the model. Although these pseudo alignments can be utilized to partially alleviate the problem of ignoring future context, they are computationally expensive to obtain.

In this paper, we propose a self-supervised model specifically designed for the word alignment task, namely MASK-ALIGN. Our model parallelly masks out each target token and recovers it conditioned on the source and other target tokens. Figure 1 shows an example where the target token "Tokyo" is masked out and re-predicted. Intuitively, as all source tokens except "Tokio" can find their counterparts on the target side, "Tokio" should be aligned to the masked token. Based on this intuition, we assume that the source token contributing most to recovering a masked target token should be aligned to that target token. Compared with NMT-based methods, MASK-ALIGN is able to take full advantage of bidirectional context on the target side and hopefully achieves higher alignment quality. We also introduce an attention variant called *leaky attention* to reduce the high attention weights on specific tokens such as periods. By encouraging agreement between two directional models both for training and inference, our method consistently outperforms the state-of-the-art on four language pairs without using guided alignment loss.

2 Approach

Figure 2 shows the architecture of our model. The model predicts each target token conditioned on the source and other target tokens and generates alignments from the attention weights between source and target (Section 2.1). Specifically, our approach introduces two attention variants, *static-KV attention* and *leaky attention*, to efficiently obtain attention weights for word alignment. To better utilize attention weights from two directions, we encourage agreement between two unidirectional models during both training (Section 2.2) and inference (Section 2.3).

2.1 Modeling

Conventional unsupervised neural aligners are based on NMT models (Peter et al., 2017; Garg et al., 2019). Given a source sentence $\mathbf{x} = x_1, \ldots, x_J$ and a target sentence $\mathbf{y} = y_1, \ldots, y_I$, NMT models the probability of the target sentence conditioned on the source sentence:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{i=1}^{I} P(y_i|\mathbf{y}_{<i},\mathbf{x};\theta) \quad (1)$$

where $\mathbf{y}_{<i}$ is a partial translation. One problem of this type of approaches is that they fail to exploit the future context on the target side, which is probably helpful for word alignment.

To address this problem, we model the same conditional probability but predict each target token

y_i conditioned on the source sentence $\mathbf{x}$ and the remaining target tokens $\mathbf{y}\backslash y_i$:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{i=1}^{I} P(y_i|\mathbf{y}\backslash y_i, \mathbf{x};\theta) \quad (2)$$

This equals to masking out each y_i and then recovering it. We build our model on top of Transformer (Vaswani et al., 2017) which is the state-of-the-art sequence-to-sequence architecture. Next, we will discuss in detail the implementation of our model.

Static-KV Attention

As self-attention is fully-connected, directly computing $\prod_{i=1}^{I} P(y_i|\mathbf{y}\backslash y_i, \mathbf{x};\theta)$ with a vanilla Transformer requires I separate forward passes, in each of which only one target token is masked out and predicted. This is costly and time-consuming. Therefore, how to parallelly mask out and predict all target tokens in a single pass is important.

To do so, a major challenge is to avoid the representation of a masked token getting involved in the prediction process of itself. Inspired by Kasai et al. (2020), we modify the self-attention in the Transformer decoder to perform the forward passes concurrently. Given the word embedding $\mathbf{w}_i$ and position embedding $\mathbf{p}_i$ for target token y_i, we first separate the query inputs $\mathbf{q}_i$ from key $\mathbf{k}_i$ and value inputs $\mathbf{v}_i$ to prevent the to-be-predicted token itself from participating in the prediction:

$$\mathbf{q}_i = \mathbf{p}_i \mathbf{W}^Q \quad (3)$$
$$\mathbf{k}_i = (\mathbf{w}_i + \mathbf{p}_i)\mathbf{W}^K \quad (4)$$
$$\mathbf{v}_i = (\mathbf{w}_i + \mathbf{p}_i)\mathbf{W}^V \quad (5)$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V$ are parameter matrices. The hidden representation $\mathbf{h}_i$ for y_i is computed by attending to keys and values, $\mathbf{K}_{\neq i}$ and $\mathbf{V}_{\neq i}$, that correspond to the remaining tokens $\mathbf{y}\backslash y_i$:

$$\mathbf{h}_i = \text{Attention}(\mathbf{q}_i, \mathbf{K}_{\neq i}, \mathbf{V}_{\neq i}) \quad (6)$$
$$\mathbf{K}_{\neq i} = \text{Concat}(\{\mathbf{k}_m | m \neq i\}) \quad (7)$$
$$\mathbf{V}_{\neq i} = \text{Concat}(\{\mathbf{v}_m | m \neq i\}) \quad (8)$$

In this way, we ensure that $\mathbf{h}_i$ is isolated from the word embedding $\mathbf{w}_i$ in a single decoder layer. However, there exists a problem of information leakage if we update the key and value inputs for each position across decoder layers since they will contain the representation of each position from previous layers. Therefore, we keep the key and value inputs unchanged and only update the query inputs

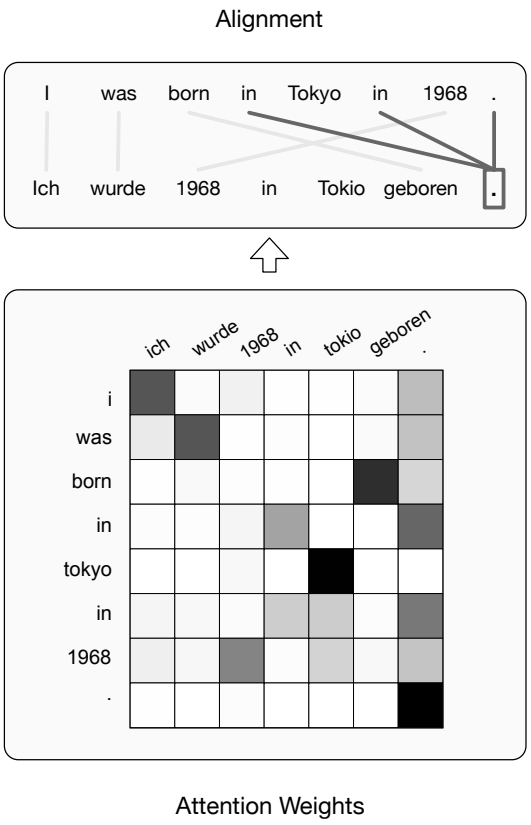


Figure 3: An example of inducing alignments from attention weights where the source token "." has high attention weights. The two "in"s in the target sentence are wrongly aligned to "." because of the high attention weights on it.

to avoid information leakage:

$$\mathbf{h}_i^l = \text{Attention}(\mathbf{q}_i^l, \mathbf{K}_{\neq i}, \mathbf{V}_{\neq i}) \quad (9)$$
$$\mathbf{q}_i^l = \mathbf{h}_i^{l-1} \mathbf{W}^Q \quad (10)$$

where $\mathbf{q}_i^l$ and $\mathbf{h}_i^l$ denote the query inputs and hidden states for y_i in the l-th layer, respectively. $\mathbf{h}_i^0$ is initialized with $\mathbf{p}_i$. We name this variant of attention the **static-KV attention**. By static-KV, we mean the keys and values are unchanged across different layers in our approach. Our model replaces all self-attention in the decoder with static-KV attention.

Leaky Attention

Extracting alignments from vanilla cross-attention often suffers from the high attention weights on some specific source tokens such as periods, [EOS], or other high frequency tokens (see Figure 3). This is similar to the "garbage collectors" effect (Moore, 2004) in statistical aligners, where a source token is aligned to too many target tokens. Hereinafter, we will refer to these tokens as *collectors*. As a result of such effect, many target tokens (e.g., the

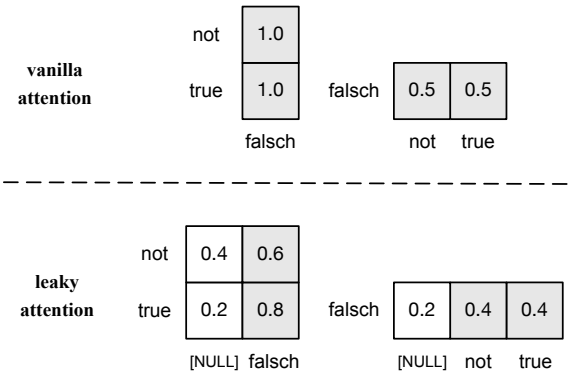


Figure 4: An illustrative example of the attention weights from two directional models using vanilla and leaky attention. Leaky attention provides a leak position "[NULL]" to collect extra attention weights.

two "in"s in Figure 3) will be incorrectly aligned to the collectors according to the attention weights.

This phenomenon has been studied in previous works (Clark et al., 2019; Kobayashi et al., 2020). Kobayashi et al. (2020) show that the norms of the value vectors for the collectors are usually small, making their influence on attention outputs actually limited. We conjecture that this phenomenon is due to the incapability of NMT-based aligners to deal with tokens that have no counterparts on the other side because there is no empty (NULL) token that is widely used in statistical aligners (Brown et al., 1993; Och and Ney, 2003).

We propose to explicitly model the NULL token with an attention variant, namely **leaky attention**. As shown in Figure 4, when calculating cross-attention weights, leaky attention provides an extra "leak" position in addition to the encoder outputs. Acting as the NULL token, this leak position is expected to address the biased attention weight problem. To be specific, we parameterize the key and value vectors as $\mathbf{k}_{\text{NULL}}$ and $\mathbf{v}_{\text{NULL}}$ for the leak position in the cross-attention, and concatenate them with the transformed vectors of the encoder outputs. The attention output $\mathbf{z}_i$ is computed as follows:

$$\mathbf{z}_i = \text{Attention}(\mathbf{h}_i^L \mathbf{W}^Q, \mathbf{K}, \mathbf{V}) \quad (11)$$

$$\mathbf{K} = \text{Concat}(\mathbf{k}_{\text{NULL}}, \mathbf{H}_{\text{enc}} \mathbf{W}^K) \quad (12)$$

$$\mathbf{V} = \text{Concat}(\mathbf{v}_{\text{NULL}}, \mathbf{H}_{\text{enc}} \mathbf{W}^V) \quad (13)$$

where $\mathbf{H}_{\text{enc}}$ denotes encoder outputs. [2] We use a normal distribution with a mean of 0 and a small deviation to initialize $\mathbf{k}_{\text{NULL}}$ and $\mathbf{v}_{\text{NULL}}$ to ensure that their initial norms are rather small. When extracting alignments, we only consider the attention matrix without the leak position.

Note that leaky attention is different from adding a special token in the source sequence, which will share the same high attention weights with the existing collector instead of calibrating it (Vig and Belinkov, 2019). Our parameterized method is more flexible than Leaky-Softmax (Sabour et al., 2017) which adds an extra dimension with the value of zero to the routing logits. In Section 2.2, we will show that leaky attention is also helpful for applying agreement-based training on two directional models.

We remove the cross-attention in all but the last decoder layer. This makes the interaction between the source and target restricted in the last layer. Our experiments demonstrate that this modification improves alignment results with fewer model parameters.

2.2 Training

To better utilize the attention weights from two directions, we apply an agreement loss in the training process to improve the symmetry of our model, which has proven effective in statistical alignment models (Liang et al., 2006; Liu et al., 2015). Given a parallel sentence pair $\langle \mathbf{x}, \mathbf{y} \rangle$, we can obtain the attention weights from two different directions, denoted as $\boldsymbol{W}_{\mathbf{x}\to\mathbf{y}}$ and $\boldsymbol{W}_{\mathbf{y}\to\mathbf{x}}$. As alignment is bijective, $\boldsymbol{W}_{\mathbf{x}\to\mathbf{y}}$ is supposed to be equal to the transpose of $\boldsymbol{W}_{\mathbf{y}\to\mathbf{x}}$. We encourage this kind of symmetry through an agreement loss:

$$\mathcal{L}_a = \text{MSE}\left(\boldsymbol{W}_{\mathbf{x}\to\mathbf{y}}, \boldsymbol{W}_{\mathbf{y}\to\mathbf{x}}^\top\right) \quad (14)$$

where MSE represents the mean squared error.

For vanilla attention, $\mathcal{L}_a$ is hardly small because of the normalization constraint. As shown in Figure 4, due to the use of softmax activation, the minimal value of $\mathcal{L}_a$ is 0.25 for vanilla attention. Using leaky attention, our approach can achieve a lower agreement loss ($\mathcal{L}_a$ = 0.1) by adjusting the weights on the leak position.

However, our model may converge to a degenerate case of zero agreement loss where attention weights are all zero except for the leak position. We circumvent this case by introducing an entropy

[2] A similar attention implementation can be found in https://github.com/pytorch/fairseq/blob/master/fairseq/modules/multihead_attention.py.

loss on the attention weights:

$$\mathcal{L}_{e,\mathbf{x}\to\mathbf{y}} = -\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{J}\tilde{W}^{ij}_{\mathbf{x}\to\mathbf{y}}\log\tilde{W}_{ij} \quad (15)$$

$$\tilde{W}^{ij}_{\mathbf{x}\to\mathbf{y}} = \frac{W^{ij}_{\mathbf{x}\to\mathbf{y}}+\lambda}{\sum_j (W^{ij}_{\mathbf{x}\to\mathbf{y}}+\lambda)} \quad (16)$$

where $\tilde{W}^{ij}_{\mathbf{x}\to\mathbf{y}}$ is the renormalized attention weights and λ is a smoothing hyperparamter. Similarly, we have $\mathcal{L}_{e,\mathbf{y}\to\mathbf{x}}$ for the inverse direction.

We jointly train two directional models using the following loss:

$$\mathcal{L} = \mathcal{L}_{\mathbf{x}\to\mathbf{y}} + \mathcal{L}_{\mathbf{y}\to\mathbf{x}} + \alpha\mathcal{L}_a + \beta(\mathcal{L}_{e,\mathbf{x}\to\mathbf{y}} + \mathcal{L}_{e,\mathbf{y}\to\mathbf{x}}) \quad (17)$$

where $\mathcal{L}_{\mathbf{x}\to\mathbf{y}}$ and $\mathcal{L}_{\mathbf{y}\to\mathbf{x}}$ are NLL losses, α and β are hyperparameters.

2.3 Inference

When extracting alignments, we compute an alignment score S_{ij} for y_i and x_j as the harmonic mean of attention weights $W^{ij}_{\mathbf{x}\to\mathbf{y}}$ and $W^{ji}_{\mathbf{y}\to\mathbf{x}}$ from two directional models:

$$S_{ij} = \frac{2\,W^{ij}_{\mathbf{x}\to\mathbf{y}}\,W^{ji}_{\mathbf{y}\to\mathbf{x}}}{W^{ij}_{\mathbf{x}\to\mathbf{y}} + W^{ji}_{\mathbf{y}\to\mathbf{x}}} \quad (18)$$

We use the harmonic mean because we assume a large S_{ij} requires both $W^{ij}_{\mathbf{x}\to\mathbf{y}}$ and $W^{ji}_{\mathbf{y}\to\mathbf{x}}$ to be large. Word alignments can be induced from the alignment score matrix as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } S_{ij} \geq \tau \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where τ is a threshold.

3 Experiments

3.1 Datasets

We conducted our experiments on four public datasets: German-English (De-En), English-French (En-Fr), Romanian-English (Ro-En) and Chinese-English (Zh-En). The Chinese-English training set is from the LDC corpus that consists of 1.2M sentence pairs. For validation and testing, we used the Chinese-English alignment dataset from Liu et al. (2005)[3], which contains 450 sentence pairs for validation and 450 for testing. For other three language pairs, we followed the experimental setup in (Zenkel et al., 2019, 2020) and used the preprocessing scripts from Zenkel et al. (2019)[4]. Following Ding et al. (2019), we take the last 1000 sentences of the training data for these three datasets as validation sets. We used a joint source and target Byte Pair Encoding (BPE) (Sennrich et al., 2016) with 40k merge operations. During training, we filtered out sentences with the length of 1 to ensure the validity of the masking process.

3.2 Settings

We implemented our model based on the Transformer architecture (Vaswani et al., 2017). The encoder consists of 6 standard Transformer encoder layers. The decoder is composed of 6 layers, each of which contains static-KV attention while only the last layer is equipped with leaky attention. We set the embedding size to 512, the hidden size to 1024, and attention heads to 4. The input and output embeddings are shared for the decoder.

We trained the models with a batch size of 36K tokens. We used early stopping based on the prediction accuracy on the validation sets. We tuned the hyperparameters via grid search on the Chinese-English validation set as it contains gold word alignments. In all of our experiments, we set $\lambda = 0.05$ (Eq. (16)), $\alpha = 5$, $\beta = 1$ (Eq. (17)) and $\tau = 0.2$ (Eq. (19)). The evaluation metric is Alignment Error Rate (AER) (Och and Ney, 2000).

3.3 Baselines

We introduce the following unsupervised neural baselines besides two statistical baselines FastAlign and GIZA++:

- NAIVE-ATT (Garg et al., 2019): a method that induces alignments from cross-attention weights of the best (usually penultimate) decoder layer in a vanilla Tranformer.
- NAIVE-ATT-LAST: same as NAIVE-ATT except that only the last decoder layer performs cross-attention.
- ADDSGD (Zenkel et al., 2019): a method that adds an extra alignment layer to repredict the to-be-aligned target token.
- MTL-FULLC (Garg et al., 2019): a method that supervises an attention head with symmetrized NAIVE-ATT alignments in a multi-task learning framework.

[3] http://nlp.csai.tsinghua.edu.cn/~ly/systems/TsinghuaAligner/TsinghuaAligner.html

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

Method	Guided	De-En	En-Fr	Ro-En	Zh-En
FAST-ALIGN (Dyer et al., 2013)	N	25.7	12.1	31.8	-
GIZA++ (Och and Ney, 2003)	N	17.8	6.1	26.0	18.5
NAIVE-ATT (Garg et al., 2019)	N	31.9	18.5	32.9	28.9
NAIVE-ATT-LAST	N	28.4	17.7	32.4	26.4
ADDSGD (Zenkel et al., 2019)	N	21.2	10.0	27.6	-
MTL-FULLC (Garg et al., 2019)	N	20.2	7.7	26.0	-
BAO (Zenkel et al., 2020)	N	17.9	8.4	24.1	-
SHIFT-ATT (Chen et al., 2020)	N	17.9	6.6	23.9	20.2
MTL-FULLC-GZ (Garg et al., 2019)	Y	16.0	4.6	23.1	-
BAO-GUIDED (Zenkel et al., 2020)	Y	16.3	5.0	23.4	-
SHIFT-AET (Chen et al., 2020)	Y	15.4	4.7	21.2	17.2
MASK-ALIGN	N	**14.4**	**4.4**	**19.5**	**13.8**

Table 1: Alignment Error Rate (AER) scores on four datasets for different alignment methods. The lower AER, the better. "Guided" denotes whether the guided alignment loss is used during training. All results are symmetrized. We highlight the best results for each language pair in bold.

- BAO (Zenkel et al., 2020): an improved version of ADDSGD that extracts alignments with Bidirectional Attention Optimization.
- SHIFT-ATT (Chen et al., 2020): a method that induces alignments when the to-be-aligned tatget token is the decoder input instead of the output.

We also included three additional baselines with guided training: (1) MTL-FULLC-GZ (Garg et al., 2019) which replaces the alignment labels in MTL-FULLC with GIZA++ results, (2) BAO-GUIDED (Zenkel et al., 2020) which uses alignments from BAO for guided alignment training, (3) SHIFT-AET (Chen et al., 2020) which trains an additional alignment module with supervision from symmetrized SHIFT-ATT alignments.

3.4 Main Results

Table 1 shows the results on four datasets. Our approach significantly outperforms all statistical and neural baselines. Specifically, it improves over GIZA++ by 1.7-6.5 AER points across different language pairs without using any guided alignment loss, making it a good substitute to this commonly used statistical alignment tool. Compared to SHIFT-ATT, the best neural methods without guided training, our approach achieves a gain of 2.2-6.4 AER points with fewer parameters (as we remove some cross-attention sublayers in the decoder).

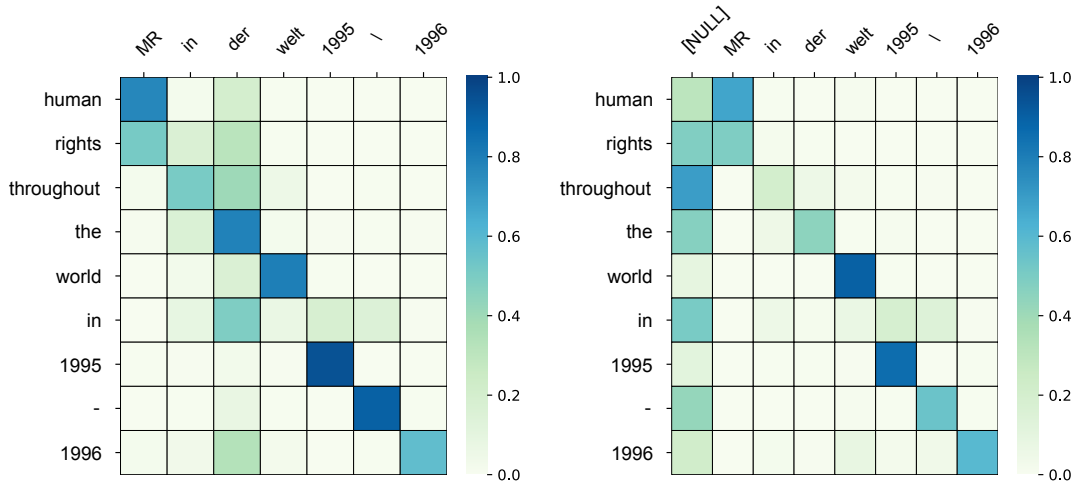
When compared with baselines using guided training, we find MASK-ALIGN still achieves substantial improvements over all methods. For example, on the Romanian-English dataset, it improves over SHIFT-AET by 1.7 AER points. Recall that our method is fully end-to-end, which does not require a time-consuming process of obtaining pseudo alignments for full training data.

Masked	Leaky	Agree	AER
×	×	×	28.4
✓	×	×	27.2
×	✓	×	28.3
×	×	✓	26.6
×	✓	✓	23.4
✓	×	✓	17.6
✓	✓	×	17.2
✓	✓	✓	**14.4**

Table 2: Ablation study on the German-English dataset. We use "Masked" to denote the masked modeling with static-KV attention in Section 2.1, "Leaky" to denote the leaky attention in Section 2.1 and "Agree" to denote the agreement-based training and inference in Sections 2.2 and 2.3.

3.5 Ablation Study

Table 2 shows the ablation results on the German-English dataset. As we can see, masked modeling seems to play a critical role since removing it will deteriorate the performance by at least 9.0 AER. We also find that leaky attention and agreement-based training and inference are both important. Removing any of them will significantly diminish



(a) Vanilla Attention (b) Leaky Attention

Figure 5: Attention weights from vanilla and leaky attention. "MR" is short for "menschenrechte", which means "human rights" in English. We use "`[NULL]`" to denote the leak position.

source sentence	`[NULL]`	`MR`	`in`	`der`	`welt`	`1995`	`\`	`1996`
vanilla attention	-	21.1	11.7	**5.2**	15.0	21.2	17.7	21.8
leaky attention	**1.9**	28.5	17.2	18.1	20.2	24.2	21.4	23.8

Table 3: Norms of the transformed value vectors of different source tokens in Figure 5. We mark the minimum norm for each variant of attention with boldface.

the performance.

3.6 Effect of Leaky Attention

Figure 5 shows the attention weights from vanilla and leaky attention and Table 3 presents the norms of the transformed value vectors of each source token for two types of attention. For vanilla attention, we can see large weights on the high frequency token "der" and the small norm of its transformed value vector. As a result, the target token "in" will be wrongly aligned to "der". While for leaky attention, we observe a similar phenomenon on the leak position "`[NULL]`", and "in" will not be aligned to any source tokens since the weights on all source tokens are small. This example shows leaky attention can effectively prevent the collector phenomenon.

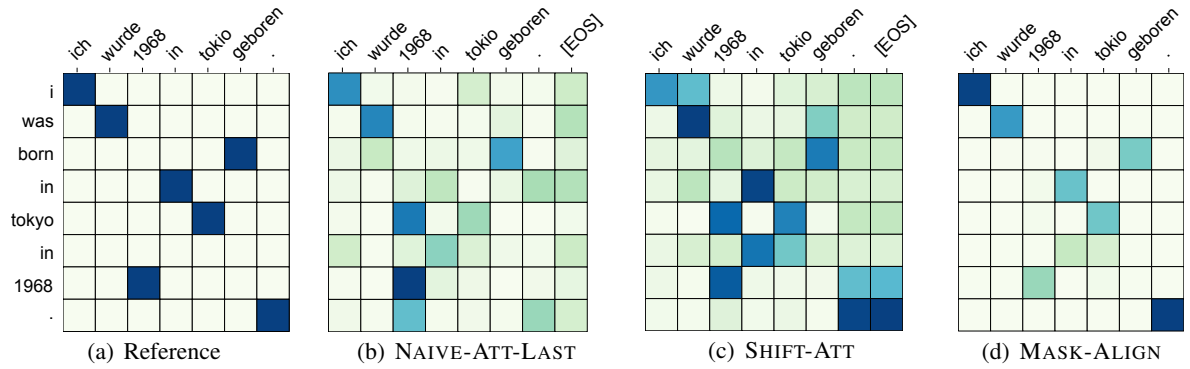
3.7 Analysis

Removing End Punctuation To further investigate the performance of leaky attention, we tested an extraction method that excludes the attention weights on the end punctuation of a source sentence. The reason behind this is that when the source sentence contains the end punctuation, it will act as the collector in most cases. Therefore removing it will alleviate the effect of collectors to a certain extent. Table 4 shows the comparison results. For vanilla attention, removing end punctuation obtains a gain of 7.7 AER points. For leaky attention, however, such extraction method brings no improvement on alignment quality. This suggests that leaky attention can effectively alleviate the problem of collectors.

Method	w/ punc.	w/o punc.
vanilla attention	27.2	17.7
leaky attention	17.2	17.4

Table 4: Comparison of AER with and without considering the attention weights on end punctuation.

Case Study Figure 6 shows the attention weights from four different models for the example in Figure 1. As we have discussed in Section 1, in this example, NMT-based methods might fail to resolve ambiguity when predicting the target token "tokyo". From the attention weight matrices, we can see that NMT-based methods (Figures 6(b) and 6(c)) indeed put high weights wrongly on "1968" in the source sentence. As for MASK-ALIGN, we can see



(a) Reference (b) NAIVE-ATT-LAST (c) SHIFT-ATT (d) MASK-ALIGN

Figure 6: Attention weights from different models for the example in Figure 1. Gold alignment is shown in (a). For target token "tokyo", NMT-based methods NAIVE-ATT-LAST (b) and SHIFT-ATT (c) assign high weights to the wrongly aligned source token "1968", while MASK-ALIGN (d) focuses on the correct source token "tokio".

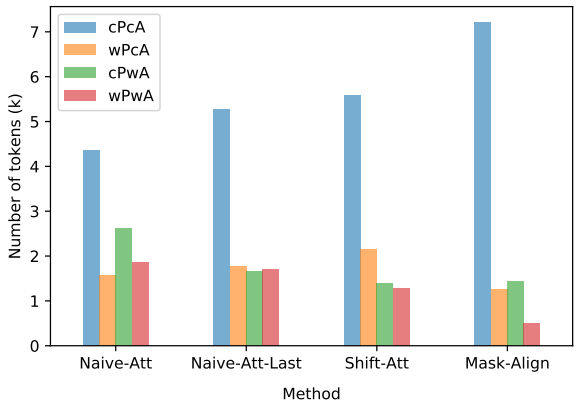


Figure 7: Relations between prediction and alignment for different methods.

that the attention weights are highly consistent with the gold alignment, showing that our method can generate sparse and accurate attention weights.

Prediction and Alignment We analyzed the relevance between the correctness of word-level prediction and alignment. We regard a word as correctly predicted if any of its subwords are correct and as correctly aligned if one of its possible alignment is matched. Figure 7 shows the results. We divide target tokens into four categories:

1. cPcA: correct prediction & correct alignment;
2. wPcA: wrong prediction & correct alignment;
3. cPwA: correct prediction & wrong alignment;
4. wPwA: wrong prediction & wrong alignment.

Compared with other methods, MASK-ALIGN significantly reduces the alignment errors caused by wrong predictions (wPwA). In addition, the number of the tokens with correct prediction but wrong alignment (cPwA) maintains at a low level, indicating that our model does not degenerate into a target masked language model despite the use of bidirectional target context.

4 Related Work

Our work is closely related to unsupervised neural word alignment. While early unsupervised neural aligners (Tamura et al., 2014; Alkhouli et al., 2016; Peter et al., 2017) failed to outperform their statistical counterparts such as FAST-ALIGN (Dyer et al., 2013) and GIZA++ (Och and Ney, 2003), recent studies have made significant progress by inducing alignments from NMT models (Garg et al., 2019; Zenkel et al., 2019, 2020; Chen et al., 2020). Our work differs from prior studies in that we design a novel self-supervised model that is capable of utilizing more target context than NMT-based models to generate high quality alignments without using guided training.

Our work is also inspired by the success of conditional masked language models (CMLMs) (Ghazvininejad et al., 2019), which have been applied to non-autoregressive machine translation. The CMLM can leverage both previous and future context on the target side for sequence-to-sequence tasks with the masking mechanism. Kasai et al. (2020) extend it with a disentangled context Transformer that predicts every target token conditioned on arbitrary context. By taking the characteristics of word alignment into consideration, we propose to use static-KV attention to achieve masking and aligning in parallel. To the best of our knowledge, this is the first work that incorporates a CMLM into alignment models.

5 Conclusion

We have presented a self-supervised neural alignment model MASK-ALIGN. Our model parallelly masks out and predicts each target token. We propose static-KV attention and leaky attention to achieve parallel computation and address the "garbage collectors" problem, respectively. Experiments show that MASK-ALIGN achieves new state-of-the-art results without using the guided alignment loss. In the future, we plan to extend our method to directly generate symmetrized alignments without leveraging the agreement between two unidirectional models.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFB0202204), National Natural Science Foundation of China (No.61925601, No. 61772302) and Huawei Noah's Ark Lab. We thank all anonymous reviewers for their valuable comments and suggestions on this work.

References

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *Association for Machine Translation in the Americas*, page 121.

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.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Ido Dagan, Kenneth Church, and Willian Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Very Large Corpora: Academic and Industrial Perspectives*.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. 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.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language 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 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention module is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing and the 10th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA. Association for Computational Linguistics.

Chunyang Liu, Yang Liu, Maosong Sun, Huanbo Luan, and Heng Yu. 2015. Generalized agreement for bidirectional word alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1828–1836, Lisbon, Portugal. Association for Computational Linguistics.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 459–466, Ann Arbor, Michigan. Association for Computational Linguistics.

Robert C. Moore. 2004. Improving IBM word alignment model 1. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 518–525, Barcelona, Spain.

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.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural automatic post-editing using prior alignment and reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain. Association for Computational Linguistics.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):27–36.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3856–3866.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

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.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz

Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

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.