Alignment-Enhanced Transformer for Constraining NMT with Pre-Specified Translations

Kai Song,[1,2] Kun Wang,[1] Heng Yu,[2] Yue Zhang,[3]
Zhongqiang Huang,[2] Weihua Luo,[2] Xiangyu Duan,[1] Min Zhang[1]
[1]Soochow University, Suzhou, China
[2]Machine Intelligence Technology Lab, Alibaba Group, Hangzhou, China
[3]School of Engineering, Westlake University, Hangzhou, China
{songkai.sk, yuheng.yh, z.huang, weihua.luowh}@alibaba-inc.com,
kwang1994@stu.suda.edu.cn, zhangyue@wias.org.cn, {xiangyuduan, minzhang}@suda.edu.cn

Abstract

We investigate the task of constraining NMT with pre-specified translations, which has practical significance for a number of research and industrial applications. Existing works impose pre-specified translations as lexical constraints during decoding, which are based on word alignments derived from target-to-source attention weights. However, multiple recent studies have found that word alignment derived from generic attention heads in the Transformer is unreliable. We address this problem by introducing a dedicated head in the multi-head Transformer architecture to capture external supervision signals. Results on five language pairs show that our method is highly effective in constraining NMT with pre-specified translations, consistently outperforming previous methods in translation quality.

1 Introduction

Neural machine translation (NMT) (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017) takes an end-to-end approach to generate translation from a source sentence, where no explicit word alignment is required during model training or decoding. NMT is less configurable and interpretable than traditional phrase-based methods (Koehn, Och, and Marcu 2003; Chiang 2007), making it difficult to incorporate external resources such as user-provided glossaries (Alkhouli, Bretschner, and Ney 2018), terminology dictionaries for specific domains (Dinu et al. 2019), and other resources. In typical industrial applications, users need to produce pre-specified translations in NMT's output (Song et al. 2019).

To this end, prior studies focus on two main approaches. The first is to use *placeholder tags* (Crego et al. 2016; Wang et al. 2017) to incorporate named-entity translations into the NMT process. Specific placeholder tags are used to substitute named entities on both the source and target sides during training. During decoding, named-entities in the source sentence are replaced with placeholder tags, which are translated into the corresponding target placeholder tags and then replaced with the translation of named-entities. The second method employs pre-specified translations to guide NMT decoding directly, taking target dictionaries as *lexical constraints* in the decoding process by imposing them on the translation output (Hokamp and Liu 2017; Post and Vilar 2018).

The first method works well for named-entities, but placeholder tags are too generic for general words and the translation quality can be negatively affected due to the loss of word meaning. Recent research finds that the second method outperforms the placeholder approach (Song et al. 2019). It retains the identity of a matched dictionary translation and models its interaction with the decoding context. However, the decoding constraints do not explicitly estimate the correlation between the source and target sides of the pre-specified translations, as the lexical constraints are only imposed on the target side. It may hurt translation fidelity because there is little or no consideration for which source word would produce the target constraint word. One way to improve the constraint method is to consider the aligned source words when generating target constraint words, as in (Alkhouli, Bretschner, and Ney 2018; Crego et al. 2016; Hasler et al. 2018) that rely on explicit word alignment based on the attention mechanism in NMT. However, it is well known that the multi-head attention mechanism employed in the Transformer architecture is ill-suited for deriving accurate word alignment (Li et al. 2019; Ding, Xu, and Koehn 2019).

In this paper, we consider the method of employing pre-specified translations to guide NMT decoding directly. Inspired by LISA (Strubell et al. 2018), which shows that external supervision improves a Transformer-based model by bringing in syntactic information, we use a dedicated attention head to learn the word alignment based on supervision from external alignment signals. Compared to existing methods (Crego et al. 2016; Hasler et al. 2018), which extract alignment information based on weights from generic attention heads, we use the dedicated attention head to learn explicit word alignment and use it to guide the constrained decoding process.

On 13 development and test sets across five different language pairs, our method achieves an average improvement of 4.48 BLEU score when using simulated pre-specified translations, in comparison to 3.51 BLEU score of the lexical constraint approach (Post and Vilar 2018).

2 Related Work

Hokamp and Liu (2017) propose modified beam search algorithm, the grid beam search, which takes target-side pre-specified translations as lexical constraints during beam search. One problem with this method is that translation fidelity is not explicitly considered, as there is no indication between the matched source words and each pre-specific translation. Another drawback is that the decoding speed is significantly reduced. Post and Vilar (2018) give a faster version of Hokamp and Liu (2017)'s work by using dynamic beam allocation in beam search to reduce decoding complexity.

Hasler et al. (2018) employ alignment between target-side constraints and their corresponding source words, simultaneously using finite-state machines and multi-stack (Anderson et al. 2017) decoding to guide beam search. Arthur, Neubig, and Nakamura (2016) use alignment to inject an external lexicon into the inference process to improve the translations of low-frequency content words. Chatterjee et al. (2017) enhance the NMT decoder with the ability to prioritize and adequately handle translation options of source words, which are located by making use of alignment information.

Alkhouli, Bretschner, and Ney (2018) study the quality of the alignments extracted from target-to-source attention in Transformer, and propose to improve alignment accuracy by injecting external alignment signals. Instead of using the attention weights over all source words to compute the source context, they add a special attention head whose source context is computed only over the aligned source words according to the external word alignment. In order to provide the alignment signal during decoding, a separate "self-attentive alignment model" is trained to learn source-side alignment jumps, also from the same external word alignment. Their use of a separate alignment model, however, to a large extent increases decoding complexity, requiring special treatment to optimize speed. In addition, a discrepancy between the quality of alignment used in training and decoding affects model performance negatively.

Zenkel, Wuebker, and DeNero (2019) propose to add a separate alignment layer to the Transformer architecture and learn to focus its attention weights on relevant source words for a given target word, in an unsupervised way from bilingual data without using external word alignment information. The learned word alignment from their attention layer is more accurate than that from a vanilla Transformer model.

Our work is related to both Alkhouli, Bretschner, and Ney (2018) and Zenkel, Wuebker, and DeNero (2019) in the sense that we all aim to improve alignment in the Transformer model. Our work differs from Alkhouli, Bretschner, and Ney (2018) in that we use a dedicated attention head to directly emulate target-to-source alignment behavior of an external alignment model, instead of using a separate alignment model to learn source-side alignment jumps. Zenkel, Wuebker, and DeNero (2019) aim to learn word alignment using a dedicated attention head without supervision from external word alignment. To our knowledge, we are the first to use a dedicated attention head to learn word alignment for guiding constrained decoding in NMT.

Strubell et al. (2018) present a linguistically-informed self-attention architecture, a neural network model that combines multi-head self-attention with multi-task learning across different NLP tasks. Our work is directly inspired by this method, but we introduce external word alignment information rather than syntactic information using dedicated attention heads.

3 Background

3.1 Transformer

Vaswani et al. (2017) use a self-attention network for both encoder and decoder in NMT. The encoder is composed of N stacked neural layers. For time step i in layer j, the hidden state h_i^j is calculated as follows: First, a self-attention sub-layer is employed to encode the context. Then *attention weights* are computed as scaled dot products between the current query h_i^{j-1} and all the keys $\{h_1^{j-1}, h_2^{j-1}, \cdots, h_m^{j-1}\}$, normalized with a softmax function. The context vector is then represented as *weighted sum* of the values projected from the hidden states in the previous layer, which are $\{h_1^{j-1}, h_2^{j-1}, \cdots, h_m^{j-1}\}$. The hidden states in the previous layer and the context vector are then connected by residual connection, followed by a layer normalization function (Ba, Kiros, and Hinton 2016) to produce a candidate hidden state $h_i^{j'}$. Finally, another sub-layer, a feed-forward network (FFN), is connected with $h_i^{j'}$ through a residual connection, followed by a layer normalization function to obtain the hidden state h_i^j.

The decoder is also composed of N stacked layers. For time step t in layer j, a self-attention sub-layer is calculated by employing self-attention mechanism over the hidden states in the previous target layer, which are $\{s_1^{j-1}, s_2^{j-1}, \cdots, s_{t-1}^{j-1}\}$, resulting in candidate hidden state $s_t^{j'}$. Then, a target-to-source sub-layer is inserted after the first self-attention sub-layer. In particular, $s_t^{j'}$ is taken as query (Q), and the keys (K) and values (V) are projected from the source hidden states in the last layer of the encoder. The attention weights $\{\alpha_{t,1}^j, \alpha_{t,2}^j, \cdots, \alpha_{t,m}^j\}$ are used to gain source context c_t^j, which is a weighted sum of source-side hidden states. Another candidate state $s_t^{j''}$ is calculated by employing the source context c_t^j and the candidate hidden state $s_t^{j'}$, which is produced by the first sub-layer. Finally, a last feed-forward sub-layer is connected with $s_t^{j''}$ through a residual connection, followed by a layer normalization function to obtain the hidden state s_t^j.

A softmax layer based on the decoder's last layer s_t^N is used to produce a probability distribution over the target-side vocabulary:

$$p(y_t|y_1, \cdots, y_{t-1}, \mathbf{x}) = \mathrm{softmax}(s_t^N * \mathbf{W}), \qquad (1)$$

where $\mathbf{W}$ is the learned weight matrix, $\mathbf{x}$ is the source sentence, and $\{y_1, y_2, \cdots, y_t\}$ is the target words.

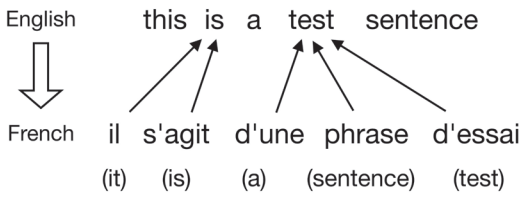


Figure 1: Word alignment derived from a vanilla Transformer model.

3.2 Alignment Extraction of Vanilla Transformer

A common and naive way to extract word alignment from Transformer is to choose the source word with the maximum attention weight towards the current target word (Crego et al. 2016; Hasler et al. 2018; Arthur, Neubig, and Nakamura 2016). In particular, an aligned source word is determined by choosing the source position which has the maximum accumulated attention weights:

$$\gamma(t) = \underset{i \in \{1,\cdots,m\}}{\text{argmax}} \frac{1}{N} \sum_{j=1}^{N} \alpha_{t,i}^{j}, \tag{2}$$

where i is the candidate aligned source-side position. For decoding step t in layer j, $\alpha_{t,i}^{j}$ is the attention weight of the i-th position in the source, which is calculated as described in Section 3.1.

As shown in Figure 1, the word alignment derived from this method are quite erroneous. This problem is also noted in (Koehn and Knowles 2017; Li et al. 2019; Ding, Xu, and Koehn 2019).

4 Method

4.1 Supervised Alignment Using a Dedicated Attention Head

Inspired by Strubell et al. (2018), we extend Transformer's multi-head self-attention architecture by adding an additional attention head, that is supervised by external alignment information. As shown in Figure 2, an additional attention head is added to the target-to-source sub-layer of each decoder layer.

In particular, at time step t in layer j, after calculating the first self-attention sub-layer's hidden state $s_t^{j'}$, two sets of attention weights are calculated in the target-to-source sub-layer: the original multi-head attention and the additional attention. Both sets of attention weights are calculated using the identical query (Q), keys (K) and values (V), but for different purposes:

1. The original multi-head attention weights $\{\alpha_{t,1}^{j}, \alpha_{t,2}^{j}, \cdots, \alpha_{t,m}^{j}\}$ over source positions $\{1, \cdots, m\}$ are used to produce candidate hidden state $s_t^{j''}$ as in the vanilla Transformer.
2. The additional attention weights $\{\beta_{t,1}^{j}, \beta_{t,2}^{j}, \cdots, \beta_{t,m}^{j}\}$ capture the target-to-source word alignment, as supervised by the external word alignment information.

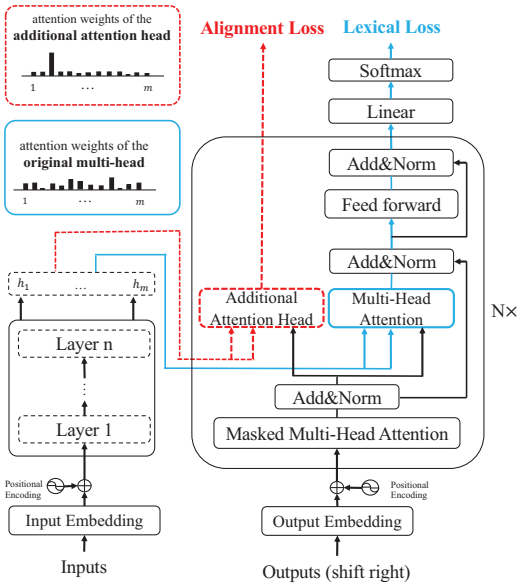


Figure 2: Additional supervised attention head.

To supervise the additional attention using the external alignment information, we introduce an alignment loss:

$$L_t^{align} = -\log \sum_{i=1}^{m} (\bar{\beta}_{t,i} \times \hat{\alpha}_{t,i}) \tag{3}$$

where $\bar{\beta}_{t,i}$ is the attention weight of the i-th source position averaged across all decoder layers:

$$\bar{\beta}_{t,i} = \frac{1}{N} \sum_{j=1}^{N} \beta_{t,i}^{j} \tag{4}$$

and $\hat{\alpha}_{t,i}$ is set to 1 only if the target word y_t is aligned to the source word x_i according to the external alignment supervision, otherwise it is 0.

The final objective function L consists of both the translation loss and the alignment loss:

$$L = \sum_{t=1}^{n} (L_t^{lexical} + \lambda * L_t^{align}) \tag{5}$$

where λ is set to 0.3 empirically, and $L_t^{lexical}$ is the original word prediction loss:

$$L_t^{lexical} = -\log(p(y_t | y_1, \cdots, y_{t-1}, \mathbf{x})) \tag{6}$$

4.2 Alignment Extraction of Alignment-Enhanced Transformer

Different from the baseline alignment extraction method described in Section 3.2, we use only the dedicated attention head to determine the aligned source position at decoding step t:

$$\gamma(t) = \underset{i \in \{1,\cdots,m\}}{\text{argmax}} \bar{\beta}_{t,i} \tag{7}$$

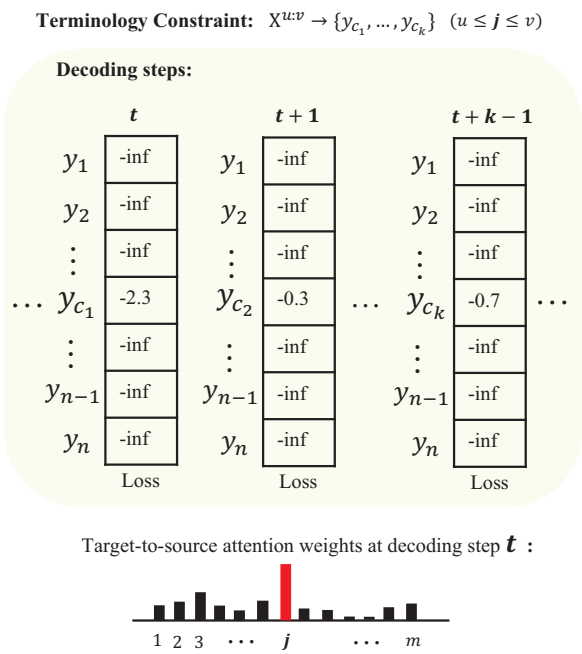


Figure 3: Dictionary-guided decoding.

where $\bar{\beta}_{t,i}$ is the average of the attention weights from all decoder layers to the i-th source position.

Different from Alkhouli, Bretschner, and Ney (2018) and Zenkel, Wuebker, and DeNero (2019), the alignment in our method is extracted by choosing the source position that has the maximum averaged attention weight produced by the additional supervised attention head, instead of the default multi-heads.

4.3 Dictionary-Guided Decoding

Alkhouli, Bretschner, and Ney (2018) describe a "dictionary-guided decoding task" as a down-stream task of leveraging Transformer's alignment, which provides an efficient way of using pre-specified translations to guide the decoding procedure. In particular, at decoding step t, if the source aligned word x_j matches a dictionary translation which should be used as a translation constraint, the decoder's output probabilities over the target-side vocabulary are reset. Infinite cost is set for all except the constrained word which is the target-side of the pre-specified translation.

Inspired by the constrained decoding algorithm used in (Hokamp and Liu 2017; Post and Vilar 2018)'s work, we extend Alkhouli, Bretschner, and Ney (2018)'s method to enable the utilization of the pre-specified translation which contains multi-words or single word that can be split into multiple sub-words. The decoding process is shown in Figure 3. Considering a pre-specified translation that matches several contiguous tokens in the source sentence:

$$x^{u:v} \rightarrow \{y_{c_1}, \cdots, y_{c_k}\}, \tag{8}$$

where $x^{u:v}$ denotes source-side tokens from position u to position v, $\{y_{c_1}, \cdots, y_{c_k}\}$ is the provided translation consisting of k tokens of target vocabulary. In decoding step t, if the aligned source position j is inside (u, v), each of the following k decoding steps will be constrained. In particular, for each decoding step r ($t \le r < t+k$), the probability distribution over target vocabulary will be reset to make target token $y_{c_{r-t+1}}$ the maximum one. In addition, for decoding steps s ($s \ge t + k$), if any of the source positions inside $(u : v)$ is aligned, the above operation will not be repeatedly applied and the decoding procedure is the same as the standard NMT decoding.

Since the only difference between dictionary-guided decoding and standard NMT decoding is the adjustment operation on the loss of each word in target vocabulary, no additional calculation is required, so time and memory consumption are not affected.

5 Experiments

We use an in-house re-implementation of Transformer (Vaswani et al. 2017), similar to Google's Tensor2Tensor. We test our method on five language pairs: English to Romanian (En-Ro), English to German (En-De), English to Russian (En-Ru), English to French (En-Fr) and Chinese to English (Ch-En). BLEU[1] (Papineni et al. 2002) and alignment error rate (AER)[2] (Och and Ney 2000) are used for the evaluation of translation quality and alignment quality, respectively.

5.1 Data

Our training corpora are taken from the WMT news translation task. In particular, the training corpora of En-De and En-Ro are taken from WMT2014 and WMT2016, respectively. Training corpora for En-Ru, En-Fr and Ch-En are taken from WMT2018. To validate our method on large scale data, corpora from both real bilingual texts and synthetic back-translation (Sennrich, Haddow, and Birch 2015a) are used for these three language pairs. The synthetic corpus is translated from "NewsCommonCrawl", which can be obtained from the WMT task. The number of training sentence pairs is 0.6M, 4.5M, 14M, 38M and 25M for En-Ro, En-De, En-Ru, En-Fr and Ch-En, respectively.

To directly evaluate alignment extraction accuracy, we use two hand aligned, publicly available alignment test sets for En-Ro[3] and En-De[4]. The two test sets contain 480 sentences and 250 sentences for En-Ro and En-De, respectively.

5.2 Experimental Settings

BPE (Sennrich, Haddow, and Birch 2015b) is used in all experiments, where the number of merge operations is set to 30K for En-Ru and Ch-En, and 50K for En-Ro, En-De and En-Fr. We use six self-attention layers for both the encoder and the decoder. The embedding size and the hidden size are set to 512. Eight heads are used for the multi-head self-attention architecture. The feed-forward layer has 2,048 cells and ReLU (Krizhevsky, Sutskever, and Hinton

[1] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl
[2] https://github.com/lilt/alignment-scripts/blob/master/scripts
[3] https://www-i6.informatik.rwth-aachen.de/goldAlignment/
[4] http://web.eecs.umich.edu//~mihalcea/wpt/index.html

2012) is used as the activation function. Adam (Kingma and Ba 2014) is used for training; warmup steps are set to 16,000; the learning rate is 0.0003. We use label smoothing (Junczys-Dowmunt, Dwojak, and Sennrich 2016) with a confidence score of 0.9, and all the drop-out (Gal and Ghahramani 2016) probabilities are set to 0.1. The vocabulary size is set to 30K for Ch-En and En-Ru, 50K for En-Ro, En-Fr and Ch-En.

Alignment Supervision. We align the bilingual training corpora with FastAlign[5] (Dyer, Chahuneau, and Smith 2013) for all language pairs. For En-Ro and En-De, we additionally use word alignments produced by GIZA[6] (Och and Ney 2003) to compare the effect of the quality in alignment supervision. Both FastAlign and GIZA are used with default settings. The final alignment is performed by symmetrizing the alignments of both forward and backward directions with the grow-diag-final heuristic. All the training corpora are in sub-word format. We add a special token "eos" to the end of both source and target sentences and assume that they are aligned to each other.

Gold Pre-Specified Translations. In practice, pre-specified translations can be provided by customers or through user feedback, which contain one identified translation for specified source segment. To simulate pre-specified translations for different test sets, previous works (Hasler et al. 2018; Alkhouli, Bretschner, and Ney 2018; Post and Vilar 2018; Song et al. 2019) obtain dictionary entries by extracting translation pairs from a test set and its reference. While these dictionary entries are always correct as measured by the reference, some of them can already be generated by the baseline systems, and thus are not useful for measuring the effectiveness of dictionary-guided decoding. Moreover, some dictionary entries can be generated by one of the baseline systems but not the other, making it unfair to compare different approaches to dictionary-based decoding. In order to address these two issues, we propose to extract pre-specified translations that can correct translation errors for all baseline systems, using the method described below.

Given a source sentence $\{x_1, x_2, ..., x_m\}$ and its translation output $\{o_1^{\mathrm{A}}, o_2^{\mathrm{A}}, ..., o_r^{\mathrm{A}}\}$ produced from system A, the word alignment between them is obtained by FastAlign. The alignment between the source sentence and the reference $\{y_1, y_2, ..., y_n\}$ is also obtained by FastAlign. Given a source word x_i, if its aligned target word o_t^{A} is different from the aligned reference word y_j, a candidate pre-specified translation (x_i, y_j) belongs to system A is constructed for source word x_i in the sentence. Additionally, for a source phrase $x^{u:v}(u \leq i < v)$ and a target phrase $y^{p:q}(p < q)$, p and q denote to the beginning and the end position in the target sentence, and u and v denote to the beginning and the end position in the source sentence. According to the phrase extraction conditions described in (Koehn et al. 2007), if $y^{p:q}$ is satisfied to be the phrase translation of $x^{u:v}$, a pre-specified translation $(x^{u:v}, y^{p:q})$ is obtained, which is phrase-level and contains a mistranslated source word. The maximum number of tokens on each side of pre-specified translations is set to 3.

For a given source sentence, the final gold dictionary entries which are used by all the systems in our experiments are obtained by taking the *intersection* of different systems' pre-specified translations. We randomly select up to four pre-specified translations per source sentence. The pre-specified translations relevant to a source sentence are used for all systems, covering 6.7%, 6.64%, 6.59%, 7.36% and 7.34% of the words in the reference for En-Ro, En-De, En-Fr, En-Ru and Ch-En, respectively. The statistic is calculated on development set.

5.3 System Configurations

We compare the following systems, in which the pre-specified translations described in Section 5.2 are used.

Baseline 1: Transformer with Lexical Constraints. The algorithm in Post and Vilar (2018), which is a way of constraining NMT with pre-specified translations, is re-implemented in our Transformer. Target-sides of pre-specified translations are used as lexical constraints, which are imposed on the translation during decoding.

Baseline 2: Vanilla Transformer with Dictionary-Guided Decoding. We use vanilla Transformer with dictionary-guided decoding described in Section 4.3 as another baseline. During decoding, aligned source words are found by using the alignment extraction method described in Section 3.2.

Our System: Alignment-Enhanced Transformer with Dictionary-Guided Decoding. For all the language pairs, we use FastAlign to obtain the alignment, that is used for supervision during training. We initialize parameters of the alignment-enhanced Transformer with the pre-trained vanilla Transformer model. The hidden size of the additional attention is set to 512. During decoding, instead of using the regular attention, the alignment is extracted from the dedicated attention head by using the method described in Section 4.2.

5.4 Results

When *using* pre-specified translations described in 5.2, the vanilla Transformer with dictionary-guided decoding does not outperform the lexical constraint method in the five language pairs in 12 out of all the 13 test sets. In contrast, alignment-enhanced Transformer is better than the lexical constraint method in 12 test sets out of all the 13 test sets for all five language pairs. On En-Fr, the alignment-enhanced Transformer with dictionary-guided decoding achieves an average gain of 4.09 BLEU when using pre-specified translations, the average gain is 1.47 BLEU higher than the lexical constraint method and 1.96 BLEU higher than the vanilla Transformer with dictionary-guided decoding.

Without using any pre-specified translations, the alignment-enhanced Transformer achieves approximately the same BLEU scores as the vanilla Transformer. As shown in Tables 1 and 2, there is no significant difference in translation quality between the alignment-enhanced Transformer and the vanilla model. This result shows that our method retains the original translation quality when no constraint is used.

[5]https://github.com/clab/fast_align

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

Systems	En-Ro			En-De			En-Fr		
	dev2016†	test2016	△	test2013†	test2014	△	dev2015†	test2015	△
Transformer	25.16	22.96	-	26.04	26.18	-	31.91	37.47	-
+ dict. guid.	28.68	26.37	+3.47	28.69	28.09	+2.11	33.99	39.64	+2.13
+ lexi. cons.	29.30	27.66	+4.42	27.88	28.56	+2.28	34.48	40.14	+2.62
Align. Enhan.	25.12	22.98	-0.01	26.15	26.11	+0.02	31.96	37.47	+0.03
+ dict. guid.	**29.39**	**28.17**	**+4.72**	**30.94**	**31.19**	**+4.96**	**35.89**	**41.67**	**+4.09**

Table 1: BLEU scores of En-Ro, En-De and En-Fr. "**Transformer**" is our in-house vanilla Transformer baseline. "**Align. Enhan.**" denotes alignment-enhanced Transformer, which is our proposed method. "**dict. guid.**" denotes dictionary-guided decoding. "**lexi. cons.**" denotes lexical constraint decoding (Post and Vilar 2018). Pre-specified translations described in Section 5.2 are used. † denotes the development set.

Systems	Ch-En				En-Ru				
	dev2017†	test2017	test2018	△	test15†	test16	test17	test18	△
Transformer	19.08	20.68	20.07	-	33.53	32.12	36.73	32.12	-
+ dict. guid.	20.32	21.60	21.35	+1.15	35.93	34.36	38.94	34.17	+2.22
+ lexi. cons.	22.69	23.35	**23.45**	+3.22	37.85	36.65	41.31	36.39	+4.42
Align. Enhan.	19.13	20.82	19.73	-0.05	33.60	32.16	36.55	32.13	-0.02
+ dict. guid.	**22.73**	**23.94**	23.43	**+3.42**	**39.14**	**37.11**	**41.68**	**36.99**	**+5.10**

Table 2: BLEU scores of Ch-En and En-Ru. System descriptions are same with Table 1. Our vanilla Transformer implementation is heavily optimized, which can be inferred from the BLEU scores of "Transformer" across various public test sets.

Systems	En-Ro	En-De
FastAlign	40.1	27.2
GIZA	28.8	18.2
Vanilla Transformer	60.7	75.7
Super. FastAlign	48.7	27.4
Super. GIZA	36.2	25.3

Table 3: Alignment error rate (%) of different systems. "Transformer" denotes vanilla Transformer. "Super. FastAlign" and "Super. GIZA" denote alignment-enhanced model supervised with alignment produced by FastAlign and GIZA, respectively.

Alignment Error Rate. To compare alignment accuracy of the alignment-enhanced Transformer with the vanilla baseline, we evaluate alignment error rate on the two alignment test sets described in Section 5.1. Since the training corpus and the test sets are all represented in sub-word units, if any source sub-word unit is matched to one target-side sub-word unit, the corresponding source word and target-side word are supposed to be aligned.

As shown in Table 3, the alignment derived from the vanilla Transformer is far from being accurate, with 60.7% and 75.7% AER on En-Ro and En-De, respectively. The alignment-enhanced Transformer significantly reduces the alignment error rate to 36.2% and 25.3% respectively.

Effect of Better Supervision. We compare the AER of the alignment-enhanced Transformer trained with different supervision signals on two language pairs, En-Ro and En-De. As shown in Table 3, GIZA can generate better alignment than FastAlign. As a result, the alignment-enhanced Transformer trained with GIZA alignment supervision derives more accurate alignment than the model trained with FastAlign alignment. The alignment derived from Transformer has each target word aligned to only one source word. This is different from GIZA or FastAlign, which allows different heuristics to produce more complex alignments. In addition, the alignment extracted from the alignment-enhanced Transformer supervised with GIZA outperform the alignment generated by FastAlign. Table 4 shows the impact made by alignment of different qualities. Better alignment supervision leads to better alignment extraction, which results in better constrained translation[7]. We observe that better alignment leads to better translation quality in dictionary-guided outputs, demonstrating the usefulness of improved alignment supervision.

En-Ro	dev2016	test2016
Vanilla Transformer	29.63	27.36
Super. FastAlign	30.75	29.46
Super. GIZA	31.77	30.37

Table 4: BLEU scores of different systems on En-Ro when using user-provided dictionary entries. System descriptions can be found in caption of Table 3.

Sample Outputs. Figure 4 compares translations of different systems when using dictionary entries. Given the source sentence "*xiongdi*(brothers) *lia*(both) *fouren*(denied) *mousha*(murder)", the vanilla Transformer fails to translate "*lia*" and "*fouren*" adequately. When constraining NMT with pre-specified translations "(*lia*, both)" and "(*fouren*, de-

[7]Pre-specified translations used in this experiment are extracted according to the three systems using the method described in Section 5.2, and is not same as the pre-specified translations used in the experiments shown in Table 1.

Source sentence:	兄弟 俩 否认 谋杀。 (*xiongdi*) (*lia*) (*fouren*) (*mousha*)
Reference:	**both** brothers **denied** the killing .
Vanilla Transformer:	**the** brothers **deny** the murder .
Transformer + dic. sugg. :	**the** brothers **denied** the murder .
Align. Enhan. + dic. sugg. :	**both** brothers **denied** killing .
dictionary entries:	**(俩, both), (否认, denied)**

Figure 4: Translation sample of different systems. System descriptions are given in the caption of Table 1. The characters in the brackets under each source word are Hanyu Pinyin, which is an official romanization system for Chinese.

System	EnRo	EnDe	EnFr	ChEn	EnRu
Transformer	77.00	56.41	49.54	38.57	48.01
Align. Enhan.	85.17	81.59	82.49	69.75	80.64

Table 5: Constrain success rate (%) of vanilla Transformer and alignment-enhanced Transformer trained with supervision obtained from FastAlign.

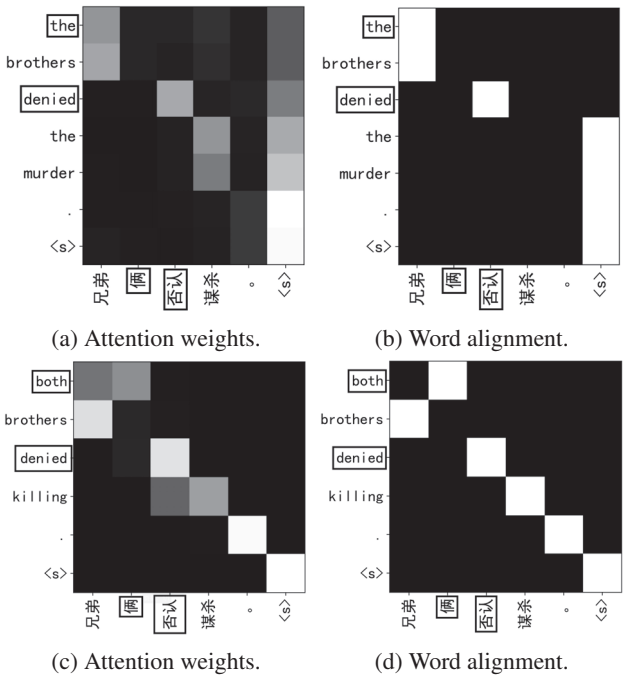
nied)", different methods lead to different outcomes.

The vanilla Transformer with dictionary-guided decoding can correct the translation of "*fouren*" with the dictionary word "denied". During decoding, the source word "*fouren*" is successfully aligned by using the alignment extraction described in Section 3.2, which matches the dictionary entries. The original word prediction probability over the target vocabulary is changed to make the provided translation "denied" surface in the final translation. However, this system fails to correct the translation of "*lia*" with the provided translation "both", because of an error in the vanilla Transformer's alignment extraction procedure.

Averaged attention weights calculated using Equation 2 are shown in Figure 5a. By using the method described in Section 3.2, the extracted alignment is shown in Figure 5b. The source word "*lia*" is not aligned to any target word, which is a frequent phenomenon in the vanilla Transformer (Li et al. 2019; Ding, Xu, and Koehn 2019). As a result, the dictionary entry "(*lia*, both)" is not used during the vanilla Transformer's decoding.

The alignment-enhanced Transformer with dictionary-guided decoding can correct both the translation of "*lia*" and the translation of "*fouren*". During decoding, the attention weights of the alignment-enhanced Transformer calculated using Equation 7 are shown in Figure 5c, leading to the extracted word alignment in Figure 5d. The source words "*lia*" and "*fouren*" are both successfully aligned during decoding, the pre-specified translations "both" and "denied" are adopted in the final translation.

Constrain Success Rate. Better alignments result in better constrain success rate. Constrain success refers to the percentage of the pre-specified translations being correctly produced in the output (Song et al. 2019), which relies heavily on the successful alignment between source word and certain target word during dictionary-guided decoding. Table 5 gives a comparison between the vanilla Transformer and



(a) Attention weights. (b) Word alignment.

(c) Attention weights. (d) Word alignment.

Figure 5: (a) and (c) are the attention weights of vanilla Transformer and alignment-enhanced Transformer, respectively. (b) and (d) are the word alignment derived from (a) and (c), respectively.

the alignment-enhanced Transformer. The latter improves the constrain success rate on all the five language pairs. By contrast, few pre-specified translations can successfully take effect in the vanilla Transformer's output because of its alignment errors. As a result, the vanilla Transformer with dictionary-guided decoding does not outperform the lexical constraint methods, as shown in Table 1 and 2.

6 Conclusion

We investigated a conceptually simple but empirically effective approach for leveraging pre-specified translations in NMT, which is a common practice in many industrial applications. Given a Transformer baseline model, an additional attention head is supervised during training, and is used to derive better word alignment, leading to improvement in dictionary-guided decoding. Results on extensive experiments show consistent improvements over two state-of-the-art techniques.

Acknowledgments

We thank the anonymous reviewers for their detailed and constructed comments. Min Zhang and Yue Zhang are the corresponding authors. The research work is supported by the National Natural Science Foundation of China (Grant No. 61525205) and the Alibaba Group through Alibaba Innovative Research Program. Thanks to Niyu Ge for the guidance in writing.

References

Alkhouli, T.; Bretschner, G.; and Ney, H. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2017. Guided open vocabulary image captioning with constrained beam search. In *EMNLP*.

Arthur, P.; Neubig, G.; and Nakamura, S. 2016. Incorporating discrete translation lexicons into neural machine translation. In *EMNLP*.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chatterjee, R.; Negri, M.; Turchi, M.; Federico, M.; Specia, L.; and Blain, F. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, 157–168.

Chiang, D. 2007. Hierarchical Phrase-based Translation. *Computational Linguistics* 33(2):201–228.

Crego, J.; Kim, J.; Klein, G.; Rebollo, A.; Yang, K.; Senellart, J.; Akhanov, E.; Brunelle, P.; Coquard, A.; Deng, Y.; et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Ding, S.; Xu, H.; and Koehn, P. 2019. Saliency-driven word alignment interpretation for neural machine translation. *WMT*.

Dinu, G.; Mathur, P.; Federico, M.; and Al-Onaizan, Y. 2019. Training neural machine translation to apply terminology constraints. In *ACL*.

Dyer, C.; Chahuneau, V.; and Smith, N. A. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Gal, Y., and Ghahramani, Z. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*.

Hasler, E.; De Gispert, A.; Iglesias, G.; and Byrne, B. 2018. Neural machine translation decoding with terminology constraints. In *ACL*.

Hokamp, C., and Liu, Q. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *ACL*.

Junczys-Dowmunt, M.; Dwojak, T.; and Sennrich, R. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. In *Proceedings of the First Conference on Machine Translation*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Koehn, P., and Knowles, R. 2017. Six challenges for neural machine translation. In *ACL*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.

Koehn, P.; Och, F.; and Marcu, D. 2003. Statistical phrase-based translation. In *Proc. NAACL/HLT*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Li, X.; Li, G.; Liu, L.; Meng, M.; and Shi, S. 2019. On the word alignment from neural machine translation. In *ACL*.

Och, F. J., and Ney, H. 2000. Improved statistical alignment models. In *ACL*.

Och, F. J., and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Post, M., and Vilar, D. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *ACL*.

Sennrich, R.; Haddow, B.; and Birch, A. 2015a. Improving neural machine translation models with monolingual data. *Computer Science*.

Sennrich, R.; Haddow, B.; and Birch, A. 2015b. Neural machine translation of rare words with subword units. In *ACL*.

Song, K.; Zhang, Y.; Yu, H.; Luo, W.; Wang, K.; and Zhang, M. 2019. Code-switching for enhancing nmt with pre-specified translation. In *NAACL*.

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wang, Y.; Cheng, S.; Jiang, L.; Yang, J.; Chen, W.; Li, M.; Shi, L.; Wang, Y.; and Yang, H. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*, 410–415.

Zenkel, T.; Wuebker, J.; and DeNero, J. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.