On the Word Alignment from Neural Machine Translation*

Xintong Li[1], Guanlin Li[2], Lemao Liu[3], Max Meng[1], Shuming Shi[3]

[1]The Chinese University of Hong Kong
[2]Harbin Institute of Technology [3]Tencent AI Lab
{znculee, epsilonlee.green}@gmail.com
{redmondliu, shumingshi}@tencent.com max.meng@cuhk.edu.hk

Abstract

Prior researches suggest that neural machine translation (NMT) captures word alignment through its attention mechanism, however, this paper finds attention may almost fail to capture word alignment for some NMT models. This paper thereby proposes two methods to induce word alignment which are general and agnostic to specific NMT models. Experiments show that both methods induce much better word alignment than attention. This paper further visualizes the translation through the word alignment induced by NMT. In particular, it analyzes the effect of alignment errors on translation errors at word level and its quantitative analysis over many testing examples consistently demonstrate that alignment errors are likely to lead to translation errors measured by different metrics.

1 Introduction

Machine translation aims at modeling the semantic equivalence between a pair of source and target sentences (Koehn, 2009), and word alignment tries to model the semantic equivalence between a pair of source and target words (Och and Ney, 2003). As a sentence consists of words, word alignment is conceptually related to machine translation and such a relation can be traced back to the birth of statistical machine translation (SMT) (Brown et al., 1993), where word alignment is the basis of SMT models and its accuracy is generally helpful to improve translation quality (Koehn et al., 2003; Liu et al., 2005).

In neural machine translation (NMT), it is also important to study word alignment, because word alignment provides natural ways to understanding black-box NMT models and analyzing their translation errors (Ding et al., 2017). Prior researches observed that word alignment is captured by NMT through attention for recurrent neural network based NMT with a single attention layer (Bahdanau et al., 2014; Mi et al., 2016; Liu et al., 2016; Li et al., 2018). Unfortunately, we surprisingly find that attention may almost fail to capture word alignment for NMT models with multiple attentional layers such as TRANSFORMER (Vaswani et al., 2017), as demonstrated in our experiments.

In this paper, we propose two methods to induce word alignment from general NMT models and answer a fundamental question that how much word alignment NMT models can learn (§ 3). The first method explicitly builds a word alignment model between a pair of source and target word representations encoded by NMT models, and then it learns additional parameters for this word alignment model with the supervision from an external aligner similar to Mi et al. (2016) and Liu et al. (2016). The second method is more intuitive and flexible: it is parameter-free and thus does not need retraining and external aligner. Its key idea is to measure the prediction difference of a target word if a source word is removed, inspired by Arras et al. (2016) and Zintgraf et al. (2017). Experiments on an advanced NMT model show that both methods achieve much better word alignment than the method by attention (§ 4.1). In addition, our experiments demonstrate that NMT captures good word alignment for those words mostly contributed from source (CFS), while their word alignment is much worse for those words mostly contributed from target (CFT). This finding offers a reason why advanced NMT models delivering excellent translation capture worse word alignment than statistical aligners in SMT, which was observed in prior researches yet without deep explanation (Tu et al., 2016; Liu et al., 2016).

Furthermore, we understand and interpret NMT from the viewpoint of word alignment induced

*Work done while X. Li interning at Tencent AI Lab. L. Liu is the corresponding author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1293–1303
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

from NMT (§ 4.2). Unlike existing researches on interpreting NMT by accessing few examples as case study (Ding et al., 2017; Alvarez-Melis and Jaakkola, 2017), we aim to provide quantitatively analysis for interpreting NMT by accessing many testing examples, which makes our findings more general. To this end, we firstly compare the effects of both approaches to interpreting NMT and find the prediction difference is better for understanding NMT. Consequently, we propose to quantitatively analyze the translation errors by using alignment from prediction difference. Since it is far from trivial to measure the translation errors at the word level, we design experiments by using two metrics to detect translation errors. Our empirical results consistently show that wrong alignment is more likely to induce the translation errors meanwhile right alignment favors to encourage the translation quality. Our analysis further suggest that word alignment errors for CFS words are responsible for translation errors in some extent.

This paper makes the two-fold contributions:

- It systematically studies word alignment from NMT and proposes two approaches to induce word alignment which are agnostic to specific NMT models.
- It understands NMT from the viewpoint of word alignment and investigates the effect of alignment errors on translation errors via quantitative analysis over many testing examples.

2 Preliminaries

2.1 Neural Machine Translation

Given a source sentence $\mathbf{x} = \langle \mathrm{x}_1, \cdots, \mathrm{x}_{|\mathbf{x}|} \rangle$ and a target sentence $\mathbf{y} = \langle \mathrm{y}_1, \cdots, \mathrm{y}_{|\mathbf{y}|} \rangle$, NMT aims at maximizing the following conditional probabilities: [1]

$$\begin{aligned} P(\mathbf{y} \mid \mathbf{x}) &= \prod_{i=1}^{|\mathbf{y}|} P(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}) \\ &= \prod_{i=1}^{|\mathbf{y}|} P\left(\mathrm{y}_i \mid \mathrm{s}_i^L\right), \end{aligned} \tag{1}$$

where $\mathbf{y}_{<i} = \langle \mathrm{y}_1, \ldots, \mathrm{y}_{i-1} \rangle$ denotes a prefix of $\mathbf{y}$ with length $i-1$, and s_i^L is the final decoding state of y_i. Generally, the conditional distribution $P\left(\mathrm{y}_i \mid \mathrm{s}_i^L\right)$ is somehow modeled within an encoder-decoder framework. In encoding stage, the source sentence $\mathbf{x}$ is encoded as a sequence of hidden vectors $\mathbf{h}$ by an encoder according to specific NMT models, such as a multi-layer encoder consisting of recurrent neural network (RNN), convolutional neural network (CNN), or self-attention layer. In decoding stage, each decoding state in l^{th} layer s_i^l is computed as follows:

$$\mathrm{s}_i^l = f\left(\mathrm{s}_i^{l-1}, \mathbf{s}_{<i}^l, \mathrm{c}_i^l\right), \quad \mathrm{s}_i^0 = \boldsymbol{y}_i, \tag{2}$$

where $l \in \{1, \ldots, L\}$, $\boldsymbol{y}_i$ is the word embedding of word y_i, f is a general function dependent on a specific NMT model, c_i^l is a context vector in l^{th} layer, computed from $\mathbf{h}$ and $\mathbf{s}_{<i}^l$ according to different NMT models. As the dominant models, attentional NMT models define the context vector c_i^l as a weighted sum of $\mathbf{h}$, where the weight $\boldsymbol{\alpha}_i^l = g\left(\mathrm{s}_i^{l-1}, \mathbf{s}_{<i}^l, \mathbf{h}\right)$ is defined by a similarity function. Due to the space limitation, we refer readers to Bahdanau et al. (2014), Gehring et al. (2017) and Vaswani et al. (2017) for the details on the definitions of f and g.

2.2 Alignment by Attention

Since the attention weight $\boldsymbol{\alpha}_{i,j}^l$ measures the similarity between s_i^{l-1} and h_j, it has been widely used to evaluate the word alignment between y_i and x_j (Bahdanau et al., 2014; Ghader and Monz, 2017). Once an attentional NMT model has been trained, one can easily extract word alignment $\boldsymbol{A}$ from the attention weight $\boldsymbol{\alpha}$ according to the style of maximum a posterior strategy (MAP) as follows:

$$\boldsymbol{A}_{i,j}(\boldsymbol{\alpha}) = \begin{cases} 1 & j = \arg\max_{j'} \boldsymbol{\alpha}_{i,j'} \\ 0 & o/w \end{cases}, \tag{3}$$

where $\boldsymbol{A}_{i,j} = 1$ indicates y_i aligns to x_j. For NMT models with multiple attentional heads attentional layers as in Vaswani et al. (2017), we sum all attention weights with respect to all heads to a single $\boldsymbol{\alpha}$ before MAP in equation 3.

3 Methods to Inducing Word Alignment

Although attention might obtain some word alignment as described in previous section, it is unknown whether NMT models contain more word alignment information than that obtained by attention. In addition, the method using attention is useful to induce word alignment for attentional

[1] Throughout this paper, bold font such as $\mathbf{x}$ denotes a sequence while regular font such as x denotes an element which may be a scalar x, vector $\boldsymbol{x}$ or matrix $\boldsymbol{X}$.

NMT models, whereas it is useless for general NMT models. In this section, in order to induce word alignment from general NMT models, we propose two different methods, which are agnostic to specific NMT models.

3.1 Alignment by Explicit Alignment Model

Given a source sentence $\mathbf{x}$, a target sentence $\mathbf{y}$, following Liu et al. (2005) and Taskar et al. (2005), we explicitly define a word alignment model as follows:

$$P\left(\mathrm{x}_j \mid \mathrm{y}_i; \boldsymbol{W}\right)=\frac{\exp \left(\delta\left(\mathrm{x}_j, \mathrm{y}_i ; \boldsymbol{W}\right)\right)}{\sum_{j^{\prime}=1}^m \exp \left(\delta\left(\mathrm{x}_{j^{\prime}}, \mathrm{y}_i ; \boldsymbol{W}\right)\right)}, \tag{4}$$

where $\delta\left(\mathrm{x}_j, \mathrm{y}_i ; \boldsymbol{W}\right)$ is a distance function parametrized by $\boldsymbol{W}$. Ideally, δ is able to include arbitrary features such as IBM model 1 similar to Liu et al. (2005). However, as our goal is not to achieve the best word alignment but to focus on that captured by an NMT model, we only consider these features completely learned in NMT. Hence, we define the

$$\delta\left(\mathrm{x}_j, \mathrm{y}_i ; \boldsymbol{W}\right)=\left(\boldsymbol{x}_j \| \boldsymbol{h}_j\right)^{\top} \boldsymbol{W}\left(\boldsymbol{y}_i \| \boldsymbol{s}_i^L\right), \tag{5}$$

where $\boldsymbol{x}_j$ and $\boldsymbol{y}_i$ are word embeddings of x_j and y_i learned in NMT, $\boldsymbol{h}_j$ is the hidden unit of x_j in the encoding network and $\boldsymbol{s}_i^L$ is the hidden unit of y_j in the decoding network, $\|$ denotes the concatenation of a pair of column vectors of dimension d, and $\boldsymbol{W}$ is a matrix of dimension $2d \times 2d$.

The explicit word alignment model is trained by maximizing the objective function with respect to the parameter matrix $\boldsymbol{W}$:

$$\max_{\boldsymbol{W}} \sum_{\forall j, i: \boldsymbol{A}_{ij}^{\mathrm{ref}}=1} \log P\left(\mathrm{x}_j \mid \mathrm{y}_i ; \boldsymbol{W}\right), \tag{6}$$

where $\boldsymbol{A}_{ij}^{\mathrm{ref}}$ is the reference alignment between x_j and y_i for a sentence pair $\mathbf{x}$ and $\mathbf{y}$. As the number of elements in $\boldsymbol{W}$ is up to one million (i.e., $(2 \times 512)^2$), it is not feasible to train it using a small dataset with gold alignment. Therefore, following Mi et al. (2016) and Liu et al. (2016), we run statistical word aligner such as FAST ALIGN (Dyer et al., 2013) on a large corpus and then employ resulting word alignment as the silver alignment $\boldsymbol{A}^{\mathrm{ref}}$ for training. Note that our goal is to quantify word alignment learned by an NMT model, and thus we only treat $\boldsymbol{W}$ as the parameter to be learned, which differs from the joint training all parameters including those from NMT models as in Mi et al. (2016) and Liu et al. (2016).

After training, one obtains the optimized $\boldsymbol{W}$ and then easily infers word alignment for a test sentence pair $\langle\mathbf{x}, \mathbf{y}\rangle$ via the MAP strategy as defined in equation 3 by setting $\boldsymbol{\alpha}_{i,j^{\prime}}=P\left(\mathrm{x}_{j^{\prime}} \mid \mathrm{y}_i ; \boldsymbol{W}\right)$.

Note that if word embeddings and hidden units learned by NMT models capture enough information for word alignment, the above method can obtain excellent word alignment. However, because the dataset for supervision in training definitely include some data intrinsic word alignment information, it is unclear how much word alignment is only from NMT models. Therefore, we propose the other method which is parameter-free and only dependent on NMT models themselves.

3.2 Alignment by Prediction Difference

The intuition to this method is that if y_i aligns to x_j, the relevance between y_i and x_j should be much higher than that between y_i and any other x_k with $k \neq j$. Therefore, the key to our method is that how to measure the relevance between y_i and x_j.

Sampling method Zintgraf et al. (2017) propose a principled method to measure the relevance between a pair of tokens in input and output. It is estimated by measuring how the prediction of y_i in the output changes if the input token x_j is unknown. Formally, the relevance between y_i and x_j for a given sentence pair $\langle\mathbf{x}, \mathbf{y}\rangle$ is defined as follows:

$$R\left(\mathrm{y}_i, \mathrm{x}_j\right)=P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}\right)-P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{\backslash j}\right), \tag{7}$$

with

$$\begin{aligned}
& P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{\backslash j}\right) \\
& =\sum_{\mathrm{x}} P\left(\mathrm{x} \mid \mathbf{y}_{<i}, \mathbf{x}_{(j, \varnothing)}\right) P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{(j, \mathrm{x})}\right) \\
& \approx \mathbb{E}_{\mathrm{x} \sim P(\mathrm{x})}\left[P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{(j, \mathrm{x})}\right)\right],
\end{aligned} \tag{8}$$

where $\mathbf{x}_{(j,\mathrm{x})}$ denotes the sequence by replacing x_j with x in $\mathbf{x}$, particularly $\mathbf{x}_{(j,\varnothing)}$ denotes the sequence by removing x_j from $\mathbf{x}$, $P(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x})$ is defined in equation 1 and $P\left(\mathrm{x} \mid \mathbf{y}_{<i}, \mathbf{x}_{(j,\varnothing)}\right)$ is approximated by the empirical distribution $P(\mathrm{x})$, which can be considered as the 1-gram language model for the source side of the training corpus. Unlike a computer vision task in Zintgraf et al. (2017), the size of source vocabulary in NMT is

up to 30000 and thus summation over this large vocabulary is challenging in computational efficiency. As a result, we only sample multiple words to approximate the expectation in equation 8 by Monte Carlo (MC) approach.

Deterministic method Inspired by the idea of dropout (Srivastava et al., 2014), we measure the relevance by disabling the connection between x_j and the encoder network in a deterministic way. Formally, $R\left(\mathrm{y}_i, \mathrm{x}_j\right)$ is directly defined via dropout effect on x_j as follows:

$$R\left(\mathrm{y}_i, \mathrm{x}_j\right)=P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}\right)-P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{(j, \mathbf{0})}\right), \tag{9}$$

where $\mathbf{x}_{(j,\mathbf{0})}$ denotes the sequence by replacing x_j with a word whose embedding is a zero vector. In this way, the computation in equation 9 is much faster than the Monte Carlo sampling approach involving multiple samples. It is worth mentioning that equation 9 resembles the Monte Carlo sampling approach with a single sample in calculation, but it is significantly better than MC with a single sample in alignment quality and is very close to MC approach with enough samples, as to be shown in our experiments.

Note that the relevance $R(\mathrm{y}_i, \mathrm{x}_j) \in [-1, 1]$, where $R(\mathrm{y}_i, \mathrm{x}_j) = 1$ means i^{th} target word is totally determined by the j^{th} source word; $R(\mathrm{y}_i, \mathrm{x}_j) = -1$ means i^{th} target word and j^{th} source word are mutual exclusive; $R(\mathrm{y}_i, \mathrm{x}_j) = 0$ means j^{th} source word do not affect generating i^{th} target word. To obtain word alignment for a given sentence pair $\langle\mathbf{x}, \mathbf{y}\rangle$, after collecting $R(\mathrm{y}_i, \mathrm{x}_j)$ one can easily infer word alignment via the MAP strategy as defined in equation 3 by setting $\boldsymbol{\alpha}_{i,j'} = R(\mathrm{y}_i, \mathrm{x}_{j'})$.

Remark The above $R(\mathrm{y}_i, \mathrm{x}_j)$ in equation 7 quantifies the relevance between a target word y_i and a source word x_j. Similarly, one can quantify the relevance between y_i and its history word y_k as follows:

$$R_o\left(\mathrm{y}_i, \mathrm{y}_k\right)=P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}\right)-P\left(\mathrm{y}_i \mid \mathbf{y}_{<i(k, \mathbf{0})}, \mathbf{x}\right), \tag{10}$$

where R_o indicates the relevance between two target words y_i and y_k with $k < i$, and $P(\mathrm{y}_i \mid \mathbf{y}_{<i(k,\mathbf{0})}, \mathbf{x})$ is obtained by disabling the connection between y_k and the decoder network, similarly to $P\left(\mathrm{y}_i \mid \mathbf{y}_{<i}, \mathbf{x}_{(j,\mathbf{0})}\right)$. Unlike $R(\mathrm{y}_i, \mathrm{x}_j)$ capturing word alignment information, $R_o(\mathrm{y}_i, \mathrm{y}_k)$ is able to capture word allocation in a target sentence and it will be used to answer a fundamental question why NMT models yields better translation yet worse word alignment compared with SMT in section of experiments.

4 Experiments

In this section, we conduct extensive experiments on ZH⇒EN and DE⇒EN translation tasks to evaluate different methods for word alignment induced from the NMT model and compare them with a statistical alignment model FAST ALIGN (Dyer et al., 2013). Then, we use the induced word alignment to understand translation errors both qualitatively and quantitatively.

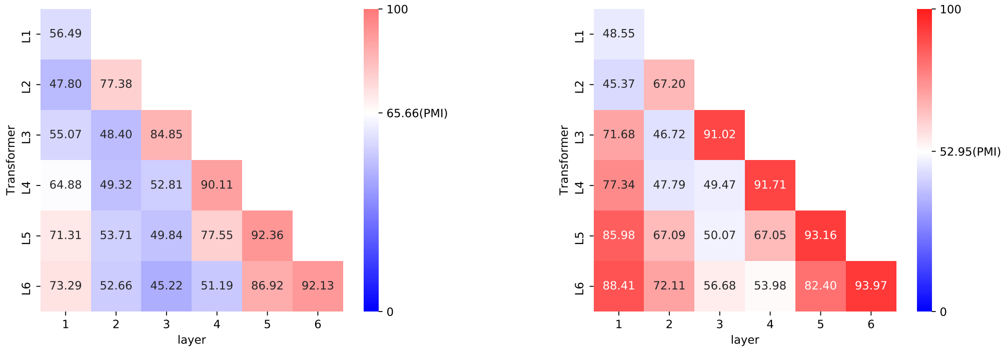
The alignment performance is evaluated by alignment error rate (AER) (Mihalcea and Pedersen, 2003; Koehn, 2009). The proposed methods are implemented on top of TRANSFORMER (Vaswani et al., 2017) which is a state-of-the-art NMT system. We report AER on NIST05 test set and RWTH data, whose reference alignment was manually annotated by experts (Liu et al., 2016; Ghader and Monz, 2017). More details on data and training these systems are described in Appendix A.

4.1 Inducing Word alignment from NMT

Attention Since the bilingual corpus intrinsically includes word alignment in some extent, word alignment by attention should be better than the data intrinsic alignment if attention indeed captures alignment. To obtain the data intrinsic word alignment, we calculate pointwise mutual information (PMI) from the bilingual corpus and then infer word alignment for each bilingual sentence by using the MAP strategy as in equation 3. [2]

It is astonishing that word alignment by attention is inconsistent for different layers of TRANSFORMER, although attention in a single layer TRANSFORMER obtains decent word alignment. Referring to Figure 1, for models more than two layers, alignment captured by attention on middle layer(s) is reasonable, but that on low or high layer is obviously worse than PMI. The possible reasons can be explained as follows. The possible functionality of lower layers might be constructing gradually better contextual representation of the word at each position as suggested in recent contextualized embedding works (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2019). So

[2] More details in Appendix B.



(a) ZH⇒EN

(b) DE⇒EN

Figure 1: AER of attention at each layer on TRANSFORMER with different number of layers. AER of PMI is shown as white. Blue and red means AER is better and worse than PMI respectively.

the AERs become better while more unambiguous representations of the corresponding word are formed. However, for higher layers the representational redundancy is accumulated (Voita et al., 2019; Michel et al., 2019) for phrases or other larger meaning spans in the input, so attention is not capturing word-to-word align but more complicated semantic correspondence.

Methods	Tasks	
	ZH⇒EN	DE⇒EN
FAST ALIGN	36.57	26.58
Attention ^{mean}	56.44	74.59
Attention ^{best}	45.22	53.98
EAM	38.88	39.25
PD	41.77	42.81

* Results are measured on TRANSFORMER-L6.

Table 1: AER of the proposed methods.

Models	TRANSFORMER					
	L1	L2	L3	L4	L5	L6
AER	54.50	47.94	40.47	38.40	38.80	38.88
BLEU	36.51	44.83	45.63	47.19	46.35	46.95

* Results are measured on ZH⇒EN task.

Table 2: EAM on translation models with different number of layer.

Explicit Alignment Model (EAM) As shown in Table 1, EAM outperforms alignment induced from attention by a large margin. However, since EAM employs silver alignment annotations from FAST ALIGN for training the additional parameters, its final AER includes contributions from both the aligned data and the model. To eliminate contribution from the data, we investigate the AERs over different pre-trained translation models with their EAMs trained on the same FAST ALIGN annotated data. We find that a stronger (higher BLEU) translation model generally obtains better alignment (lower AER). As shown in Table 2, TRANSFORMER-L6 generates much better alignment than TRANSFORMER-L1, highly correlated with their translation performances. This suggests that supervision is not enough to obtain good alignment and the hidden units learned by a translation model indeed implicitly capture alignment knowledge by learning translation. In addition, EAM can be thought as a kind of agnostic probe (Belinkov et al., 2017; Hewitt and Manning, 2019) to investigate how much alignment are implicitly learned in the hidden representations.

Prediction Difference (PD) As shown in Table 1, PD also delivers better word alignment than attention. PD can be implemented by sampling method or deterministic method. As shown in Table 3, the alignment performance of sampling method is improving as growing of the sample size, because the accuracy of Monte Carlo approach is dependent on the number of samples. And no matter what sample size is, the variance of AER is always ignorable. The reason might be that the $\arg\max$ operation in equation 3 eliminates the fluctuation of probability matrix. Although using large sample size can achieve nice alignment performance, it is costly in computation. Fortunately, the deterministic method, which employs a single zero embedding rather than embedding of random words, can also achieve nice alignment performance with the same computa-

Methods	Sampling method					Deterministic method
Sample size	1	2	4	20	50	
AER	44.92	43.30	42.42	41.83	41.73	41.77
Variance	0.004	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	N/A

* Results are measured on TRANSFORMER-L6 and ZH⇒EN task

Table 3: Comparison between sampling and deterministic methods for prediction difference.

tional. One possible reason is that using zero embedding in inference is exactly the same way as dropout in training, making the trained parameters perform well in inference. In the rest of this paper, we employ the deterministic version as the default for PD in this paper.

Alignment on CFT words It is well-known that NMT outperforms SMT a lot in translation, and thus it is natural to ask why NMT yields worse alignment than the aligner FAST ALIGN in SMT, as shown in Table 1. Because the probability of a target word typically employs the mixed contributions from both source and target sides, NMT may capture good alignment for the target words mostly contributed from source (CFS, such as content words) while bad alignment for the target words mostly contributed from target (CFT, such as function words). To this end, we divide the target words into two categories: for a given sentence pair $\langle \mathbf{x}, \mathbf{y} \rangle$, CFS and CFT are formally defined as two sets containing the target word y_i satisfies following conditions respectively,

$$\max_{\mathrm{x} \in \mathbf{x}} R(\mathrm{y}_i, \mathrm{x}) - \max_{\mathrm{y} \in \mathbf{y}_{<i}} R_o(\mathrm{y}_i, \mathrm{y}) > \epsilon,$$
$$\max_{\mathrm{y} \in \mathbf{y}_{<i}} R_o(\mathrm{y}_i, \mathrm{y}) - \max_{\mathrm{x} \in \mathbf{x}} R(\mathrm{y}_i, \mathrm{x}) > \epsilon, \tag{11}$$

where $\epsilon \in [0, 1)$ is a probability margin between CFS and CFT words.

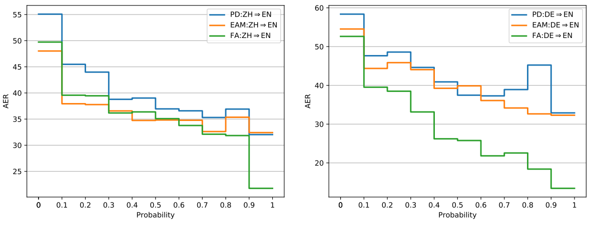
After dividing the target words into two categories of CFS and CFT words according to the criterion defined above, [3] we evaluate alignment performance for each category and the results are shown in Table 4. We find that NMT indeed captures better alignment for CFS words than the alignment for CFT words, and FAST ALIGN generates much better alignment than NMT for CFT words. Therefore, this fact indicates that CFT words are the reason why NMT generate worse alignment than FAST ALIGN.

Methods	Target Words	Tasks	
		ZH⇒EN	DE⇒EN
PD	ALL	41.77	42.81
	CFS	32.97	33.86
	CFT	63.28	65.24
EAM	ALL	38.88	39.25
	CFS	34.44	36.03
	CFT	49.73	47.34
FAST ALIGN	ALL	36.57	27.05
	CFS	31.02	22.56
	CFT	50.80	38.48

* For both tasks the ratio between CFS word count and CFT word count is about 7 : 3.

Table 4: AER of CFS and CFT words.

Confidence-binned AER Since confidence can reflect translation quality to some extent, we also use the confidence of each target word (the predictive probability) during forced decoding to group the targets into ten bins and report the AER of them in Figure 2. We can find the AER generally decreases as the probability increases. This also indicates that alignment analysis on real translation instead of ground truth may lead to more reliable conclusion since beam search always finds high confidence candidates.



(a) ZH⇒EN

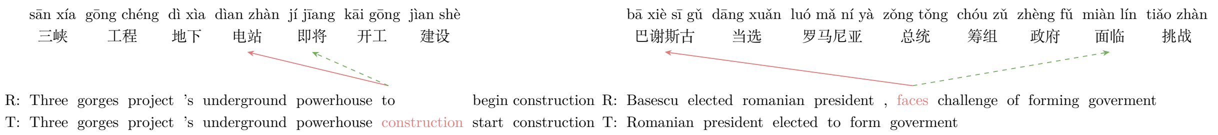
(b) DE⇒EN

Figure 2: Confidence-binned AER on the two datasets.

4.2 Understanding NMT via PD Alignment

Which method is better for understanding? Previous experiments mainly consider the alignment for the reference, and show that EAM is better at aligning a reference word to source words than PD. However, in order to better understand

[3] Without affecting main conclusions, $\epsilon = 0$ in this experiment for covering all words in analysis. Experiments with different margins are in Appendix C.


sān xía gōng chéng dì xìa dìan zhàn jí jīang kāi gōng jìan shè
三峡 工程 地下 电站 即将 开工 建设
R: Three gorges project ’s underground powerhouse to begin construction
T: Three gorges project ’s underground powerhouse construction start construction
bā xiè sī gŭ dāng xuǎn luó mǎ ní yà zǒng tǒng chóu zǔ zhèng fǔ miàn lín tiǎo zhàn
巴谢斯古 当选 罗马尼亚 总统 筹组 政府 面临 挑战
R: Basescu elected romanian president , faces challenge of forming goverment
T: Romanian president elected to form goverment

(a) Forced Decoding Error

(b) Real Decoding Error

Figure 3: Two examples of showing the translation errors caused by word alignment errors both in forced decoding and real decoding on TRANSFORMER-L6. Red arrow means wrong alignment while Green arrow means the golden alignment. red word means translation error. ‘R’ denotes reference sentence and ‘T’ denotes translation sentence.

the translation process of a NMT model, it is helpful to analyze the alignment of real translations derived from the NMT model itself. This is also in accordance with the confidence-binned observation previously. The alignment of the real translation actually provides some insight on the causal relationship among source and target words. To obtain AER on real decoding, we manually annotate word alignment of the real translations for 200 source sentences randomly selected from the ZH⇒EN test set. As shown in Table 5, PD yields better alignment for the real translation than EAM, and we even surprisingly find that its alignment performance is better than FAST ALIGN. [4] This quantitative finding demonstrates PD is better for understanding the real translation in general rather than only for some special case.

Models	AER
PD & TRANSFORMER-L6	20.44
EAM & TRANSFORMER-L6	29.77
FAST ALIGN	25.23

* Results are measured on sampled 200 sentences of ZH⇒EN task, and golden alignment for real translation are human labeled (Appendix D)

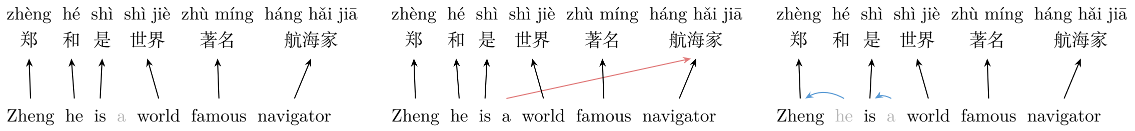
Table 5: Alignment of Real Translation.

It is worth noting that EAM does not only deliver worse word alignment for real translations than PD, but also be dangerous to understand NMT through its word alignment. The main reason is that EAM relies on an external aligned dataset with supervision from statistical word aligner FAST ALIGN, and thus the characteristic of its alignment result are similar to that of FAST ALIGN, leading to the understanding biased to FAST ALIGN. In contrast, PD only relies on prediction from a neural model to define the relevance, it has been successfully used to understand and interpret a neural model (Zintgraf et al., 2017). Therefore, in the rest of this subsection, we try to understand NMT by using PD both qualitatively and quantitatively.

Analyze translation errors in forced decoding We consider the forced decoding translation error as follows. We fix the translation history as the prefix of the reference $\mathrm{y}_{<i}$ at each timestep i and then check whether the 1-best word $\hat{\mathrm{y}}_i = \arg\max_{\mathrm{y}} P(\mathrm{y}|\mathrm{y}_{<i}, \mathrm{x})$ is exactly y_i. If $\hat{\mathrm{y}}_i \neq \mathrm{y}_i$ we say the NMT model makes an error decision at this timestep. We give a case of this kind of error in Figure 3(a). After visualizing the alignment of y_i by PD, we find that its alignment in red color is not correct compared to the ground-truth alignment in green color. As a result, the NMT model can not capture the sufficient context to accurately predict the reference word y_i and thereby generates an incorrect word ‘*construction*’.

Besides the case study, we try to quantitatively interpret that alignment errors may lead to translation errors. To this end, we divide all timesteps from the reference of the test dataset into two categories, i.e. one with right alignment and the other with wrong alignment. Then we calculate the forced decoding translation error rates for each category, i.e. the ratio between the number of timesteps making error decisions in one category and the total number of timesteps, as depicted in Table 6. From the table, it is clear that wrong alignment is more likely to cause a translation error while correct alignment is likely to make a correct translation decision. Particularly, compared with right alignment, when alignment is wrong, the forced decoding translation error rate of CFS words increases much more than CFT words (Δ). This observation indicates word alignment errors of CFS words are mainly responsible for translation errors instead of CFT words.

[4] The numbers in Table 5 are not comparable to those in Table 1 and Figure 2, because they employ different translations in the target side leading to different ground-truth alignments, which are crucial for evaluating alignment.


zhèng hé shì shì jiè zhù míng háng hǎi jiā
郑 和 是 世界 著名 航海家
Zheng he is a world famous navigator
zhèng hé shì shì jiè zhù míng háng hǎi jiā
郑 和 是 世界 著名 航海家
Zheng he is a world famous navigator
zhèng hé shì shì jiè zhù míng háng hǎi jiā
郑 和 是 世界 著名 航海家
Zheng he is a world famous navigator

(a) Gold Alignment (b) Alignment to Source Side (c) Alignment with CFS & CFT

Figure 4: An example of word alignment and translation produced by TRANSFORMER-L6. Red arrow means wrong alignment and blue arrow means the prediction is attributed to a target word. The word in light font do not align to any source word, while red word means wrong translation.

Tasks	Target Words	Right Alignment	Wrong Alignment	∆
ZH⇒EN	ALL	34.87	49.24	14.37
	CFS	35.34	53.91	18.57
	CFT	32.86	43.99	11.13
DE⇒EN	ALL	23.63	35.64	11.01
	CFS	24.21	38.25	14.04
	CFT	26.40	32.38	5.98

* Results are measured on TRANSFORMER-L6.

Table 6: Forced decoding translation error rate for CFS/CFT words with right/wrong alignment.

Analyze translation errors in real decoding Besides the forced decoding translation error, we care more about search-aware statistics in real decoding. Specifically, we identify words in the reference which are recalled through the real translation, and those unrecalled words are called real decoding translation errors defined as $\{y\} \setminus \{\hat{y}\}$, the difference between the two sets where $\{y\}$ is the set of words in y. As shown in the case in Figure 3(b), the identified translation error '*faces*' is wrongly aligned by PD to '*bā xiè sī gǔ*', which may strongly correlate to the under translation of '*miàn lín*' at the source side.

Tasks	Target Words	Right Alignment	Wrong Alignment	∆
ZH⇒EN	ALL	31.72	40.73	9.01
	CFS	31.03	41.44	10.41
	CFT	34.67	39.92	5.25
DE⇒EN	ALL	23.84	40.09	16.25
	CFS	22.31	39.04	16.73
	CFT	30.53	41.40	10.87

* Results are measured on TRANSFORMER-L6.

Table 7: Real decoding translation error rate for CFS/CFT words with right/wrong alignment.

For quantitative analysis, the same as the forced decoding, we split all target words into two parts, i.e. right alignment and wrong alignment, and then we evaluate the real decoding translation error rate for each of them via $\sum_i |\{y^i\} \setminus \{\hat{y}^i\}| / \sum_i |\{y^i\}|$. As shown in Table 7, there is an obvious gap between the real decoding translation error of right alignment and wrong alignment, which shows alignment errors have adverse effect on translation quality. For CFS and CFT words, Table 7 demonstrates that alignment errors cause decreasing of translation quality for both sets. Same as forced decoding, the real decoding translation error are also mainly attributed to CFS words. This suggests improving the ability of learning word alignment for CFS words is potential to improve translation quality for neural machine translation.

Interpret Translation via CFT Alignment As the translation error has been shown related to the alignment error, the translation success can also be understood by word alignment. Previous research (Ding et al., 2017; Alvarez-Melis and Jaakkola, 2017) have attempted to interpret the decision-making of translation by aligning target words to source words. However, there is nonignorable amount of translated target words are mostly contributed from target side instead of source side.

As shown in Figure 4(a), as a functional word, '*a*' should not be aligned to any source word. However, in Figure 4(b) PD incorrectly aligned '*a*' to '*háng hǎi jiā*' by only considering the contributions from the source side, and this leads to a misunderstanding for why '*a*' is translated. Fortunately, according to equation 11, PD is good at distinguishing where the contributions come from for both source and target sides. As shown in Figure 4(c), considering alignment of words in CFS, '*a*' is superbly not aligned to any source word because it belongs to CFT and should be aligned to '*is*', which explains why NMT correctly translates '*a*'.

Although the ambiguous Chinese word '*hé*' mostly means '*and*', TRANSFORMER is able to translate it perfectly as a given name '*hé*' as shown in Figure 4(c). [5] The main reason is that NMT captures the context of the surname '*zheng*' by PD over target side besides the context of '*hé*' by PD over source side, thanks to its more powerful language model effect.

5 Related Work

In NMT, there are many notable researches which mention word alignment captured by attention in some extent. For example, Bahdanau et al. (2014) is the first work to show word alignment examples by using attention in an NMT model. Tu et al. (2016) quantitatively evaluate word alignment captured by attention and find that its quality is much worse than statistical word aligners. Motivated by this finding, Chen et al. (2016), Mi et al. (2016) and Liu et al. (2016) improve attention with the supervision from silver alignment results obtained by statistical aligners, in the hope that the improved attention leads to better word alignment and translation quality consequently. More recently, there are also works (Alkhouli et al., 2018) that directly model the alignment and use it to sharpen the attention to bias translation. Despite the close relation between word alignment and attention, Koehn and Knowles (2017) and Ghader and Monz (2017) discuss the differences between word alignment and attention in NMT. Most of these works study word alignment for the same kind of NMT models with a single attention layer. One of our contribution is that we propose model-agnostic methods to study word alignment in a general way which deliver better word alignment quality than attention method. Moreover, for the first time, we further understand NMT through alignment and particularly quantify the effect of alignment errors on translation errors for NMT.

The prediction difference method in this paper actually provides an avenue to understand and interpret neural machine translation models. Therefore, it is closely related to many works on visualizing and interpreting neural networks (Lei et al., 2016; Bach et al., 2015; Zintgraf et al., 2017). Indeed, our method is inherited from (Zintgraf et al., 2017), and our advantage is that it is computationally efficient particularly for those tasks with a large vocabulary. In sequence-to-sequence tasks, Ding et al. (2017) focus on model interpretability by modeling how influence propagates across hidden units in networks, which is often too restrictive and challenging to achieve as argued by Alvarez-Melis and Jaakkola (2017). And instead, Alvarez-Melis and Jaakkola (2017) concentrate on prediction interpretability with only oracle access to the model generating the prediction. To achieve this effect, they propose a casual learning framework to measure the relevance between a pair of source and target words. Our method belongs to the type of prediction interpretability similar to Alvarez-Melis and Jaakkola (2017), but ours is a unified and parameter-free method rather than a pipeline and parameter-dependent one. In addition, both Ding et al. (2017) and Alvarez-Melis and Jaakkola (2017) qualitatively demonstrate interpretability by showing some sentences, while we exhibit the interpretability by *quantitatively* analyzing all sentences in a test set.

6 Conclusions and Future Work

This paper systematically studies the word alignment from NMT. It firstly reveals that attention may not capture word alignment for an NMT model with multiple attentional layers. Therefore, it proposes two methods (explicit model and prediction difference) to acquire word alignment which are agnostic to specific NMT models. Then it suggests prediction difference is better for understanding NMT and visualizes NMT from word alignment induced by prediction difference. In particular, it quantitatively analyzes that alignment errors which are likely to lead to translation errors at word level measured by different metrics. In the future, we believe more work on improving CFS alignment is potential to improve translation quality, and we will investigate on using source context and target history context in a more robust manner for better predicting CFS and CFT words.

Acknowledgements

We would like to thank all the anonymous reviewers for their valuable suggestions. This research was supported by Tencent AI Lab, Hong Kong RGC GRF grant # 14200618, Hong Kong ITC ITSP Tier 2 grant # ITS/105/18FP and Shenzhen Science and Technology Innovations project JCYJ20170413161616162.

[5] It is interesting that SMT (MOSES) incorrectly translates this word into '*and*' in our preliminary experiment.

References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *WMT*.

David Alvarez-Melis and Tommi S Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *arXiv preprint arXiv:1707.01943*.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in nlp. *arXiv preprint arXiv:1606.07298*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

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

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James R. Glass. 2017. What do neural machine translation models learn about morphology? In *ACL*.

Peter F Brown, Vincent J Della Pietra, Stephen A 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. In *Proceedings of AMTA*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

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.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? *arXiv preprint arXiv:1710.03348*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

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

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Workshop on Neural Machine Translation*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL-HLT*, pages 48–54.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *In Proceedings of EMNLP*.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1380–1390.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING*.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of ACL*, pages 459–466.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of EMNLP*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *arXiv preprint arXiv:1905.10650*.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond-Volume 3*, pages 1–10. 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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matthew Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05.

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.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.