Attention is Not Only a Weight: Analyzing Transformers with Vector Norms

Goro Kobayashi[1] **Tatsuki Kuribayashi**[1,2] **Sho Yokoi**[1,3] **Kentaro Inui**[1,3]
[1] Tohoku University [2] Langsmith Inc. [3] RIKEN
{goro.koba, kuribayashi, yokoi, inui}@ecei.tohoku.ac.jp

Abstract

Attention is a key component of Transformers, which have recently achieved considerable success in natural language processing. Hence, attention is being extensively studied to investigate various linguistic capabilities of Transformers, focusing on analyzing the parallels between *attention weights* and specific linguistic phenomena. This paper shows that attention weights alone are only one of the two factors that determine the output of attention and proposes a norm-based analysis that incorporates the second factor, the norm of the transformed input vectors. The findings of our norm-based analyses of BERT and a Transformer-based neural machine translation system include the following: (i) contrary to previous studies, BERT pays poor attention to special tokens, and (ii) reasonable word alignment can be extracted from attention mechanisms of Transformer. These findings provide insights into the inner workings of Transformers.

1 Introduction

Transformers (Vaswani et al., 2017; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Lan et al., 2020) have improved the state-of-the-art in a wide range of natural language processing tasks. The success of the models has not yet been sufficiently explained; hence, substantial research has focused on assessing the linguistic capabilities of these models (Rogers et al., 2020; Clark et al., 2019).

One of the main features of Transformers is that they utilize an attention mechanism without the use of recurrent or convolutional layers. The attention mechanism computes an output vector by accumulating relevant information from a sequence of input vectors. Specifically, it assigns attention weights (i.e., relevance) to each input, and sums up input vectors based on their weights. The analysis of correlations between attention weights and various linguistic phenomena (i.e., *weight-based analysis*) is a prominent research area (Clark et al., 2019; Kovaleva et al., 2019; Reif et al., 2019; Lin et al., 2019; Mareček and Rosa, 2019; Htut et al., 2019; Raganato and Tiedemann, 2018; Tang et al., 2018).

This paper first shows that weight-based analysis is insufficient to analyze the attention mechanism. Weight-based analysis is a common approach to analyze the attention mechanism by simply tracking attention weights. The attention mechanism can be expressed as a *weighted* sum of *linearly transformed vectors* (Section 2.2); however, the effect of transformed vectors in weight-based analysis is ignored. We propose a *norm-based analysis* that considers the previously ignored factors (Section 3). In this analysis, we measure the norms (lengths) of the vectors that were summed to compute the output vector of the attention mechanism.

Using the norm-based analysis of BERT (Section 4), we interpreted the internal workings of the model in more detail than when weight-based analysis was used. For example, the weight-based analysis (Clark et al., 2019; Kovaleva et al., 2019) reports that specific tokens, such as periods, commas, and special tokens (e.g., separator token; `[SEP]`), tend to have high attention weights. However, our norm-based analysis found that the information collected from vectors corresponding to special tokens was considerably lesser than that reported in the weight-based analysis, and the large attention weights of these vectors were canceled by other factors. Additionally, we found that BERT controlled the levels of contribution from frequent, less informative words by controlling the norms of their vectors.

In the analysis of a Transformer-based NMT system (Section 5), we reinvestigated how accurate word alignment can be extracted from the

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7057–7075,
November 16–20, 2020.

source-target attention. The weight-based results of Li et al. (2019), Ding et al. (2019), and Zenkel et al. (2019) have empirically shown that word alignments induced by the source-target attention of the Transformer-based NMT systems are noisy. Our experiments show that more accurate alignments can be extracted by focusing on the vector norms.

The contributions of this study are as follows:

- We propose a novel method of analyzing an attention mechanism based on vector norms (norm-based analysis). The method considers attention weights and previously ignored factors, i.e., the norm of the transformed vector.
- Our norm-based analysis of BERT reveals that (i) the attention mechanisms pay considerably lesser attention to special tokens than to observations that are solely based on attention weights (weight-based analysis), and (ii) the attention mechanisms tend to discount frequent words.
- Our norm-based analysis of a Transformer-based NMT system reveals that reasonable word alignment can be extracted from source-target attention, in contrast to the previous results of the weight-based analysis.

The codes of our experiments are publicly available.[1]

2 Background

2.1 Attention mechanism

Attention is a core component of Transformers, which consist of several layers, each containing multiple attentions ("heads"). We focused on analyzing the inner workings of these heads.

As illustrated in Figure 1, each attention head gathers relevant information from the input vectors. A vector is updated by vector transformations, attention weights, and a summation of vectors. Mathematically, attention computes each output vector $\boldsymbol{y}_i \in \mathbb{R}^d$ from the corresponding pre-update vector $\widetilde{\boldsymbol{y}}_i \in \mathbb{R}^d$ and a sequence of input vectors $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$:

$$\boldsymbol{y}_i = \left(\sum_{j=1}^{n} \alpha_{i,j} \boldsymbol{v}(\boldsymbol{x}_j) \right) \boldsymbol{W}^O \tag{1}$$

$$\alpha_{i,j} := \underset{\boldsymbol{x}_j \in \mathcal{X}}{\mathrm{softmax}} \left(\frac{\boldsymbol{q}(\widetilde{\boldsymbol{y}}_i) \boldsymbol{k}(\boldsymbol{x}_j)^\top}{\sqrt{d'}} \right) \in \mathbb{R}, \tag{2}$$

where $\alpha_{i,j}$ is the attention weight assigned to the token x_j for computing y_i, and $\boldsymbol{q}(\cdot)$, $\boldsymbol{k}(\cdot)$, and $\boldsymbol{v}(\cdot)$ are the query, key, and value transformations, respectively.

$$\boldsymbol{q}(\widetilde{\boldsymbol{y}}_i) := \widetilde{\boldsymbol{y}}_i \boldsymbol{W}^Q + \boldsymbol{b}^Q \quad \left(\boldsymbol{W}^Q \in \mathbb{R}^{d \times d'}, \boldsymbol{b}^Q \in \mathbb{R}^{d'} \right)$$
$$\boldsymbol{k}(\boldsymbol{x}_j) := \boldsymbol{x}_j \boldsymbol{W}^K + \boldsymbol{b}^K \quad \left(\boldsymbol{W}^K \in \mathbb{R}^{d \times d'}, \boldsymbol{b}^K \in \mathbb{R}^{d'} \right)$$
$$\boldsymbol{v}(\boldsymbol{x}_j) := \boldsymbol{x}_j \boldsymbol{W}^V + \boldsymbol{b}^V \quad \left(\boldsymbol{W}^V \in \mathbb{R}^{d \times d'}, \boldsymbol{b}^V \in \mathbb{R}^{d'} \right).$$

Attention gathers value vectors $\boldsymbol{v}(\boldsymbol{x}_j)$ based on attention weights and then, applies matrix multiplication $\boldsymbol{W}^O \in \mathbb{R}^{d' \times d}$ (Figure 1). [2] Boldface letters such as $\boldsymbol{x}$ denote row (not column) vectors, following the notations in Vaswani et al. (2017).

In self-attention, the input vectors $\mathcal{X}$ and the pre-update vector $\widetilde{\boldsymbol{y}}_i$ are previous layer's output representations. In source-target attention, $\mathcal{X}$ corresponds to the representations of the encoder, and vector $\widetilde{\boldsymbol{y}}_i$ (and updated vector $\boldsymbol{y}_i$) corresponds to the vector of the i-th input token of the decoder.

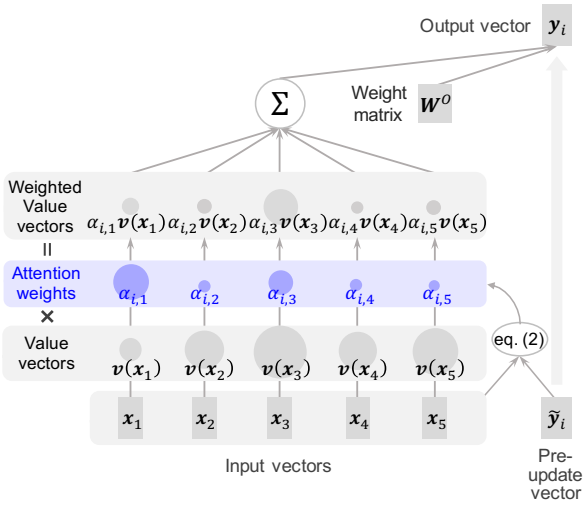


Figure 1: Overview of attention mechanism in Transformers. Sizes of the colored circles illustrate the value of the scalar or the norm of the corresponding vector.

2.2 Attention is a weighted sum of vectors

With a simple reformulation, one can observe that the attention mechanism computes the weighted sum of the transformed input vectors. Because of the linearity of the matrix product, we can rewrite Equation 1 as

$$\boldsymbol{y}_i = \sum_{j=1}^{n} \alpha_{i,j} f(\boldsymbol{x}_j) \tag{3}$$

[1] https://github.com/gorokoba560/norm-analysis-of-transformer

[2] Whether bias $\boldsymbol{b}$ is added to calculate query, key, and value vectors depends on the implementation. $\boldsymbol{W}^O \in \mathbb{R}^{d' \times d}$ in Equation 1 corresponds to the part of $\boldsymbol{W}^O \in \mathbb{R}^{hd' \times d}$ that was introduced in Vaswani et al. (2017) which is applied to each head; where h is the number of heads, and $hd' = d$ holds.

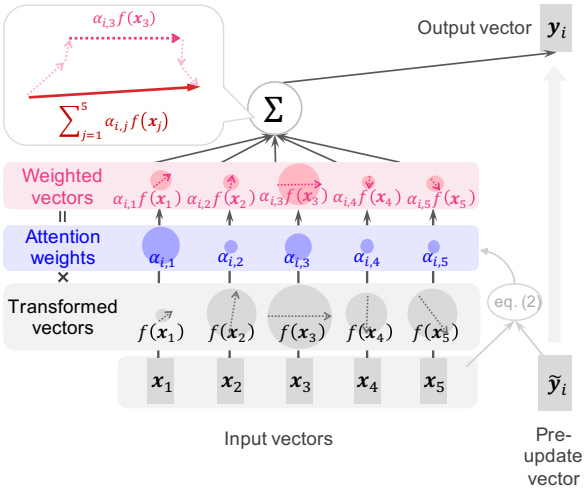


Figure 2: Overview of attention mechanism based on Equation 3. It computes the output vector by summing the weighted vectors; vectors with larger norms have higher contributions. Sizes of the colored circles illustrate the value of the scalar or the norm of the corresponding vector.

$$f(\boldsymbol{x}) := \left(\boldsymbol{x}\boldsymbol{W}^V + \boldsymbol{b}^V\right)\boldsymbol{W}^O. \tag{4}$$

Equation 3 shows that the attention mechanism first transforms each input vector $\boldsymbol{x}$ to generate $f(\boldsymbol{x})$; computes attention weights α; and then compute the *sum* $\alpha f(\boldsymbol{x})$ (see Figure 2).

2.3 Problems encountered in weight-based analysis

The attention mechanism has been designed to update representations by gathering relevant information from the input vectors. Prior studies have analyzed attention, focusing on attention weights, to ascertain which input vectors contribute (weight-based analysis) (Clark et al., 2019; Kovaleva et al., 2019; Reif et al., 2019; Lin et al., 2019; Mareček and Rosa, 2019; Htut et al., 2019; Raganato and Tiedemann, 2018; Tang et al., 2018).

Analyses solely based on attention weight are based on the assumption that the larger the attention weight of an input vector, the higher its contribution to the output. However, this assumption disregards the magnitudes of the transformed vectors. The problem encountered when neglecting the effect of $f(\boldsymbol{x}_j)$ is illustrated in Figure 2. The transformed vector $f(\boldsymbol{x}_1)$ for input $\boldsymbol{x}_1$ is assumed to be very small ($\|f(\boldsymbol{x}_1)\| \approx 0$), while its attention weight $\alpha_{i,1}$ is considerably large. Note that the small $\alpha_{i,1}f(\boldsymbol{x}_1)$ contributes a little to the output vector $\boldsymbol{y}_i$ because $\boldsymbol{y}_i$ is the sum of $\alpha f(\boldsymbol{x})$, where a larger vector contributes more to the output. Conversely, the large $\alpha_{i,3}f(\boldsymbol{x}_3)$ dominates the output $\boldsymbol{y}_i$. Therefore, in this case, only considering the attention weight may lead to a wrong interpretation of the high contribution of input vector $\boldsymbol{x}_1$ to output $\boldsymbol{y}_i$. Nevertheless, $\boldsymbol{x}_1$ hardly has any effect on $\boldsymbol{y}_i$.

Analyses based on attention weights have not provided clear results in some cases. For example, Clark et al. (2019) reported that input vectors for separator tokens `[SEP]` tend to receive remarkably large attention weights in BERT, while changing the magnitudes of these weights does not affect the masked-token prediction of BERT. Such results can be attributed to the aforementioned issue of focusing only on attention weights.

3 Proposal: norm as a degree of attention

As described in Section 2.3, analyzing the attention mechanism with only attention weights neglects the effect of the transformed vector $f(\boldsymbol{x}_j)$, which has a significant impact as we discussed later.

Herein, we propose the measurement of the *norm of the weighted transformed vector* $\|\alpha f(\boldsymbol{x})\|$, given by Equation 3, to analyze the attention mechanism behavior.[3] Unlike in previous studies, we analyzed the behaviors of the norms, $\|\alpha f(\boldsymbol{x})\|$ and $\|f(\boldsymbol{x})\|$, and α to gain more in-depth insights into the functioning of attention. The proposed method of analyzing the attention mechanism is called **norm-based analysis** and the method that solely analyzes the attention weights is called **weight-based analysis**.

In Sections 4 and 5, we provide insights into the working of Transformers using norm-based analysis. Appendix A explains that our norm-based analysis can also be effectively applied to an entire multi-head attention mechanism.

4 Experiments: BERT

First, we show that the previously ignored transformed-vector norm affects the analysis of attention in BERT (Section 4.1). Applying our norm-based analysis, we re-examine the previous reports on BERT obtained by weight-based analysis (Section 4.2). Next, we demonstrate the previously overlooked properties of BERT (Section 4.3).

[3] We use the standard Euclidean norm.

Head	μ	σ	CV	Max	Min
Layer 2–Head 4 (max CV)	4.26	1.59	**0.37**	12.66	0.96
Layer 2–Head 7 (min CV)	4.00	0.50	**0.12**	6.15	1.35
Average	5.15	1.17	**0.22**	-	-

Table 1: Mean (μ), standard deviation (σ), coefficient of variance (CV), and maximum and minimum values of $\|f(\boldsymbol{x})\|$. In the last row, the former three are averaged over all the heads.

General settings: Following the previous studies (Clark et al., 2019; Kovaleva et al., 2019; Reif et al., 2019; Lin et al., 2019; Htut et al., 2019), we used the pre-trained BERT-base[4], with 12 layers, each containing 12 attention heads. We used the data provided by Clark et al. (2019) for the analysis.[5] The data contains 992 sequences extracted from Wikipedia, where each sequence consists of two consecutive paragraphs, in the form of: `[CLS] paragraph1 [SEP] paragraph2 [SEP]`. Each sequence consists of up to 128 tokens, with an average of 122 tokens.

4.1 Does $f(\boldsymbol{x})$ have an impact?

We analyzed the coefficient of variation (CV)[6] of previously ignored effect—$\|f(\boldsymbol{x})\|$—to first demonstrate the degree to which $\|\alpha f(\boldsymbol{x})\|$ differs from weight α. We computed the CV of $\|f(\boldsymbol{x})\|$ of all the example data for each head. Table 1 shows that the average CV is 0.22. Typically, the value of the norm $\|f(\boldsymbol{x})\|$ varies from 0.78 to 1.22 times the average value of the $\|f(\boldsymbol{x})\|$. Thus, there is a difference between the weight α and $\|\alpha f(\boldsymbol{x})\|$ due to the dispersion of $\|f(\boldsymbol{x})\|$, which motivated us to consider $\|f(\boldsymbol{x})\|$ in the attention analysis. Appendix B presents the detailed results.

4.2 Re-examining previous observation

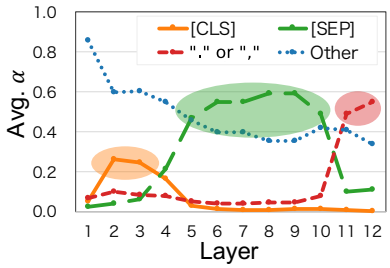
In this section, with the application of our norm-based analysis, we reinvestigate the previous observation of Clark et al. (2019); they analyzed BERT using the weight-based analysis.

Settings: First, all the data were fed into BERT. Then, the weight α and $\|\alpha f(\boldsymbol{x})\|$ were collected from each head. Following Clark et al. (2019), we report the results of the following categories: (i) `[CLS]`, (ii) `[SEP]`, (iii) periods and commas, and (iv) the other tokens. More specific descriptions of the experiments are provided in Appendix D.

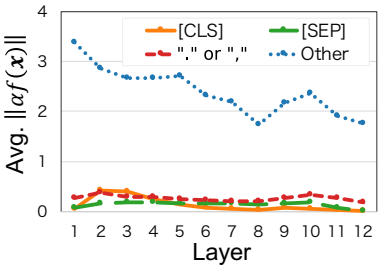
[4]We used PyTorch implementation of BERT-base (uncased) released at https://github.com/huggingface/transformers.

[5]https://github.com/clarkkev/attention-analysis

[6]*Coefficient of variation* (CV) is a standardized (scale-invariant) measure of dispersion, which is defined by the ratio of the standard deviation σ to the mean μ; CV $:= \sigma/\mu$.



(a) Weight-based analysis.



(b) Norm-based analysis.

Figure 3: Each point corresponds to averaged α or $\|\alpha f(\boldsymbol{x})\|$ on a word category in a given layer. Note that, in each layer, the sum of α among all the categories is 1. The x-axis denotes the index of the layers.

Token category	Number of vectors	Spearman's ρ
[CLS]	17,443,296	-0.34
[SEP]	34,886,592	-0.69
comma & period	182,838,528	-0.25
Others	1,944,928,224	-0.06

Table 2: Spearman rank correlation coefficient between α and $\|f(\boldsymbol{x})\|$ in each token category.

Results: The weight-based and norm-based analyses exhibited entirely different trends (Figure 3). The vectors for specific tokens—`[CLS]`, `[SEP]`, and punctuations—have remarkably large attention weights, which is consistent with the report of Clark et al. (2019). In contrast, our norm-based analysis demonstrated that the contributions of vectors corresponding to these tokens were generally small (Figure 3b). The result demonstrates that the size of the transformed vector $f(\boldsymbol{x})$ plays a considerable role in controlling the amount of information obtained from the specific tokens.

Clark et al. (2019) hypothesized that if the necessary information is not present in the input vectors, BERT assigns large weights to `[SEP]`, which appears in every input sequence, to avoid the incorporation of any additional information via at-

tention.[7] Clark et al. (2019) called this operation no-operation (no-op). However, it is unclear whether assigning large attention weights to [SEP] realizes the operation of collecting little information from the input sequence.

Our norm-based analysis demonstrates that the amount of information from the vectors corresponding to [SEP] is small (Figure 3b). This result supports the interpretation that BERT conducts "no-op," in which attention to [SEP] is considered a signal that does not collect anything. Additionally, we hope that our norm-based analysis can provide a better interpretation of other existing findings.

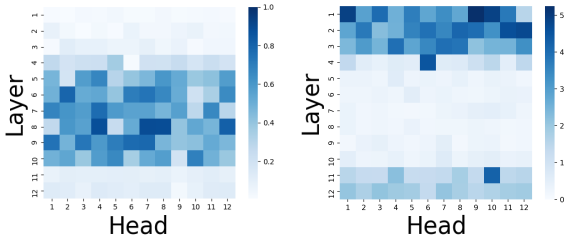
Analysis—The relationship between α and $\|f(\boldsymbol{x})\|$: It remains unclear how attention *collects only a little information* while assigning a high attention weight to a specific token, [SEP]. Here, we demonstrate an interesting trend of α and $\|f(\boldsymbol{x})\|$ cancelling each other out on the tokens.[8] Table 2 shows the Spearman rank correlation coefficient between α and $\|f(\boldsymbol{x})\|$, corresponding to the vectors in each category. The weight α and the norm $\|f(\boldsymbol{x})\|$ have a negative correlation in terms of [CLS], [SEP], periods, and commas. This cancellation manages to *collect a little information* even with large weights.

Figure 4 illustrates the contrast between α and $\|f(\boldsymbol{x})\|$ corresponding to [SEP] in each head. For most of the heads, α and $\|f(\boldsymbol{x})\|$ clearly negate the magnitudes of each other. A similar trend was observed in [CLS], periods, and commas. Conversely, no significant trend was observed in the other tokens (see Appendix D.3).

Figure 5 shows 1% randomly selected pairs of α and $\|f(\boldsymbol{x})\|$ in each word category. Even when the same weight α is assigned, $\|f(\boldsymbol{x})\|$ can vary, suggesting that α and $\|f(\boldsymbol{x})\|$ play a different roles in attention.

4.3 Relation between frequency and $\|f(\boldsymbol{x})\|$

In the previous section, we demonstrated that $\|f(\boldsymbol{x})\|$ corresponding to the specific tokens (e.g., [SEP]) is small. Based on the high frequencies[9] of these word types[10], we hypothesized that BERT controlled contributions of highly frequent, less informative words by adjusting the norm of $f(\boldsymbol{x})$.



(a) α. (b) $\|f(\boldsymbol{x})\|$.

Figure 4: The higher value of averaged α or $\|f(\boldsymbol{x})\|$ for [SEP] tokens in a given head, the darker its cell.

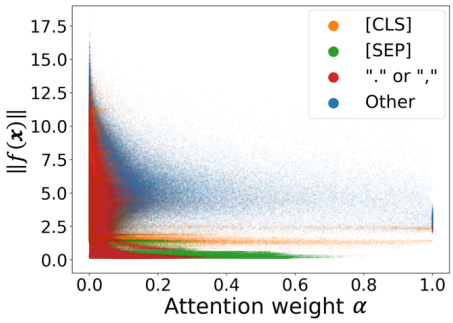


Figure 5: Relationship between α and $\|f(\boldsymbol{x})\|$. Each plot corresponds to a pair of $\alpha_{i,j}$ and $\|f(\boldsymbol{x}_j)\|$ in one of the attention heads. Each plot is colored by the word category corresponding to $\boldsymbol{x}_j$. Visualizations by category are shown in Appendix D.3.

Settings: First, all the data were fed into the model. Then, for each input token t, we collected the weight α and $\|f(\boldsymbol{x})\|$. We averaged α and $\|f(\boldsymbol{x})\|$ for all the heads for each t to analyze the trend of the entire model. Let $r(\cdot)$ be a function that returns the frequency rank of a given word.[11] We analyzed the relationship of $r(t)$ with α and $\|f(\boldsymbol{x})\|$.

Results: The Spearman rank correlation coefficient between the frequency rank $r(t)$ and $\|f(\boldsymbol{x})\|$ was 0.75, indicating a strong positive correlation. In contrast, the Spearman rank correlation coefficient did not show any correlation ($\rho = 0.06$) between $r(t)$ and α.[12] The visualizations of their relationships are shown in Appendix D.4.

These results demonstrate that the self-

[7] Note that the attention mechanism has the constraint that the sum of the attention weights becomes 1.0 (see Equation 2).

[8] Note that for any positive scalar $\lambda \in \mathbb{R}$ and vector $\boldsymbol{x} \in \mathbb{R}^d$, $\|\lambda \boldsymbol{x}\| = \lambda \|\boldsymbol{x}\|$.

[9] The frequency ranks of the words [CLS], [SEP], period, and comma, out of approximately 30,000 words, are 50, 28, 2, and 3, respectively.

[10] We call word type as "word." Each instance of a word is called "token."

[11] We counted the frequency for each word type by reproducing the training data of BERT.

[12] The Spearman rank correlation coefficient without special tokens, periods, and commas was 0.28 for the attention weights and 0.69 for the norms.

attentions in BERT reduce the information from highly frequent words by adjusting $\|f(\boldsymbol{x})\|$ and not α. This frequency-based effect is consistent with the intuition that highly frequent words, such as stop words, are unlikely to play an important role in solving the pre-training tasks (masked-token prediction and next-sentence prediction).

5 Experiments: Transformer for NMT

Additionally, we analyzed the source-target attention in a Transformer-based NMT system. One major research topic in the NMT field is whether NMT systems internally capture word alignment between source and target texts, and if so, how word alignment can be extracted from black-box NMT systems. Li et al. (2019), Ding et al. (2019), and Zenkel et al. (2019) empirically showed, using the weight-based method, that word alignment induced by the attention of the Transformer is noisy. In this section, we show the analysis of source-target attention using vector norms $\|\alpha f(\boldsymbol{x})\|$ and demonstrate that clean alignments can be extracted from the source-target attention. Word alignment can be used to provide rich information for the users of NMT systems (Ding et al., 2019).

Experimental procedure: Following Zenkel et al. (2019) and Ding et al. (2019), we trained a Transformer-based NMT system for German-to-English translation on the Europarl v7 corpus[13]. Next, we extracted word alignments from α and $\|\alpha f(\boldsymbol{x})\|$ under the force decoding setup. Finally, we evaluated the derived alignment using the alignment error rate (AER) (Och and Ney, 2000). A low AER score indicates that the extracted word alignments are close to the reference. We used the gold alignment dataset provided by Vilar et al. (2006)[14]. Experiments were performed on five random seeds, and the average AER scores were reported. The experimental settings are detailed in Appendix E.

5.1 Alignment extraction from attention

Weights or norms: A typical alignment extraction method uses attention weights (Li et al., 2019; Ding et al., 2019; Zenkel et al., 2019). Specifically, given a source-target sentence pair, $\{s_1, \ldots, s_J\}$ and $\{t_1, \ldots, t_I\}$, word alignment is estimated by calculating a source word s_j that has the highest weight when generating a target word t_i. We call this method the *weight-based alignment extraction*. In contrast, we propose a *norm-based alignment extraction* method that extracts word alignments based on $\|\alpha f(\boldsymbol{x})\|$ instead of α. Formally, in these methods, the source word s_j with the highest attention weight or norm during the generating of target word t_i is extracted as the word that is aligned with t_i:

$$\underset{s_j}{\operatorname{argmax}}\, \alpha_{i,j} \quad \text{or} \quad \underset{s_j}{\operatorname{argmax}}\, \|\alpha_{i,j} f(\boldsymbol{x}_j)\|. \tag{5}$$

In Section 5.2, following Li et al. (2019), we analyze the word alignments that we obtained from each layer by integrating H heads within the same layer:

$$\underset{s_j}{\operatorname{argmax}} \sum_{h=1}^{H} \alpha_{i,j}^h \quad \text{or} \quad \underset{s_j}{\operatorname{argmax}} \|\sum_{h=1}^{H} \alpha_{i,j}^h f^h(\boldsymbol{x}_j)\|,$$

where $f^h(\boldsymbol{x}_j)$ and $\alpha_{i,j}^h$ are the transformed vector and the attention weight at the h-th head, respectively.

Alignment with input or output word: In our preliminary experiments (Appendix E.3), we observed that the behavior of the source-target attention of the decoder differs between the earlier and later layers. As shown in Figure 6, at the time decoding the word t_{i+1} with the input t_i, attention heads in the earlier layers assign large weights or norms to s_j corresponding to the input t_i "a," whereas those in the latter layers assign large values to s_j corresponding to the output word t_{i+1} "student."

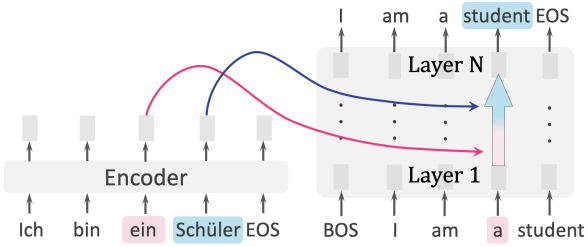


Figure 6: An example of behavior of the source-target attentions in an NMT system (German-to-English). Attentions in the earlier layers focus the source word "ein" aligned with the input word "a," while those in the latter layers focus the source word "Schüler" aligned with the output word "student."

[13] http://www.statmt.org/europarl/v7

[14] https://www-i6.informatik.rwth-aachen.de/goldAlignment/

Based on this observation, we explored two settings for investigating alignment extraction methods: *alignment with output* (AWO) and *alignment with input* (AWI). The AWO setting refers to the approach introduced in Equation 5. Specifically, alignments (s_j, t_i) were extracted by considering a source word s_j that gained the highest weight (norm) when *outputting* a particular target word t_i.

In the AWI setting, alignments (s_j, t_i) were extracted by considering a source word s_j that gained the highest weight (norm) when *inputting* the word t_i (i.e., predicting a word t_{i+1}). Formally, alignment with the AWI setting is calculated as follows:

$$\underset{s_j}{\operatorname{argmax}}\, \alpha_{i+1,j} \quad \text{or} \quad \underset{s_j}{\operatorname{argmax}}\, \|\alpha_{i+1,j} f(\boldsymbol{x}_j)\|. \tag{6}$$

5.2 Comparative experiments

We compared the quality of the alignments that were obtained by the following six methods:

- norm-based extraction with the AWO/AWI settings
- weight-based extraction with the AWO/AWI settings (Li et al., 2019; Zenkel et al., 2019; Ding et al., 2019)
- gradient-based extraction (Ding et al., 2019)
- existing word aligners (Och and Ney, 2003; Dyer et al., 2013)

We report the best and averaged AER scores across the layers. In addition, we report on the AER score at the head and the layer with the highest average $\|\alpha f(\boldsymbol{x})\|$ in the norm-based extraction.[15] The settings are detailed in Appendix E.2.

The AER scores of each method are listed in Table 3. The results show that word alignments extracted using the proposed norm-based approach are more reasonable than those extracted using the weight-based approach. Additionally, better word alignments were extracted in the AWI setting than in the AWO setting. The alignment extracted using the layer with the highest average $\|\alpha f(\boldsymbol{x})\|$ in the AWI setting is better than the gradient-based method, and competitive with one of the existing word aligners—fast_align.[16] These results show that much clearer word alignments can be extracted from a Transformer-based NMT system than the results reported by existing research.

Methods	AER	±SD
Transformer – Attention-based Approach		
— *Alignment with output* setting —		
Weight-based		
layer mean	68.4	1.0
best layer (layer 4 or 5)	47.7	1.7
Norm-based (ours)		
layer mean	62.9	0.7
best layer (layer 5)	41.4	1.4
layer with the highest average $\|\alpha f(\boldsymbol{x})\|$	83.0	1.1
head with the highest average $\|\alpha f(\boldsymbol{x})\|$	87.1	2.3
— *Alignment with input* setting —		
Weight-based		
layer mean	68.5	1.9
best layer (layer 2)	29.8	3.7
Norm-based (ours)		
layer mean	60.4	1.3
best layer (layer 2)	25.0	1.5
layer with the highest average $\|\alpha f(\boldsymbol{x})\|$	25.0	1.5
head with the highest average $\|\alpha f(\boldsymbol{x})\|$	35.5	21.0
Transformer – Gradient-based Approach		
SmoothGrad from Ding et al. (2019)	36.4	-
Word Aligner		
fast_align from Zenkel et al. (2019)	28.4	-
GIZA++ from Zenkel et al. (2019)	21.0	-

Table 3: AER scores with different methods for German-to-English translation. The closer the extracted word alignment is to the reference, the lower the AER score. The "layer mean" denotes the average of AER scores across all layers. Each value is the average of five random seeds.

The primary reason behind the differences between the results of the weight- and norm-based methods was analogous to the finding discussed in Section 4.2, while some specific tokens, such as ⟨/s⟩, the special token for the end of the sentence, tended to obtain heavy attention weights; their transformed vectors were adjusted to be smaller, as shown in Figure 7.

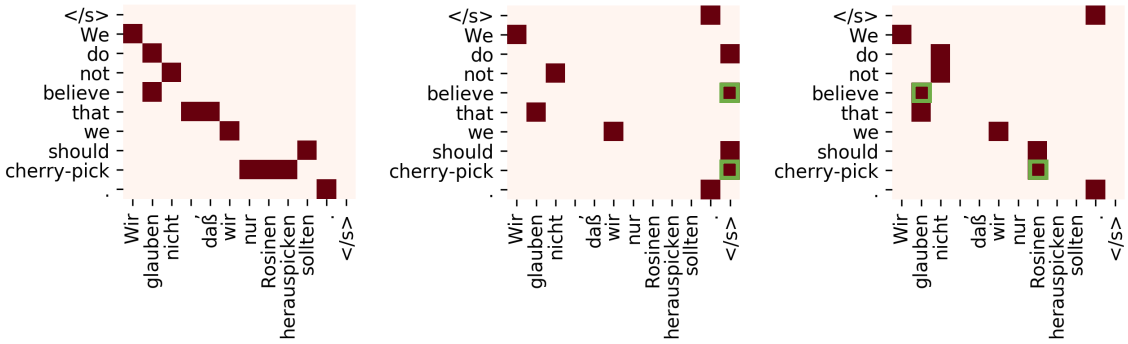
5.3 Relationship between norms and alignment quality

We further analyze the relationship between $\|\alpha f(\boldsymbol{x})\|$ and AER scores in the head-level. Figures 8a and 8b show the AER scores of the alignments obtained by the norm based extraction at each head in the AWO and AWI settings. Figure 8c shows the average of $\|\alpha f(\boldsymbol{x})\|$ at each head. The small $\|\alpha f(\boldsymbol{x})\|$ implies that α and $\|f(\boldsymbol{x})\|$ tend to cancel out in the head.

Comparing Figures 8a and 8c, the average $\|\alpha f(\boldsymbol{x})\|$ and AER scores in the AWI setting

[15] The average $\|\alpha f(\boldsymbol{x})\|$ of the layer was determined by the sum of the average $\|\alpha f(\boldsymbol{x})\|$ at each head in the layer.

[16] Even at the head with the highest average $\|\alpha f(\boldsymbol{x})\|$. Although the average score of five seeds in the AWI setting was 35.5, four seeds out of them achieved great score range from 23.6-to 25.7. The score was 77.5 for a remaining seed.



(a) Reference. (b) α. (c) $\|\alpha f(\boldsymbol{x})\|$.

Figure 7: Examples of the reference and extracted alignments using each method in layer 2 (best layer) in the AWI setting on one out of five seeds. Two misalignments in the weight-based extraction were resolved in the norm-based analysis—alignments with the green frame. Examples of the extracted alignments in all the layers are shown in Appendix E.4.

are inversely correlated (the Spearman rank and Pearson correlation coefficients are -0.44 and -0.52, respectively). This result is consistent with Table 3, where the head or the layer with the highest average $\|\alpha f(\boldsymbol{x})\|$ provides clean alignments in the AWI setting. This result suggests that Transformer-based NMT systems may rely on specific heads that align source and target tokens. This result is also consistent with the exiting reports that pruning some attention heads in Transformers does not change its performance; on the contrary, it improves the performance (Michel et al., 2019; Kovaleva et al., 2019).

In contrast, in the AWO setting (Figures 8b and 8c), such a negative correlation is not observed; rather, a positive correlation is observed (Spearman's ρ is 0.56, and the Pearson's r is 0.55). Actually, in the AWO setting, the alignments extracted from the head/layer with the highest $\|\alpha f(\boldsymbol{x})\|$ is considerably worse than those from the other settings in Table 3. Investigating the reason for these contrasting results would be our future work. In Appendix F, we also present the results of a model with a different number of heads.

6 Related work

6.1 Probing of Transformers

Transformers are used for many NLP tasks. Many studies have probed their inner workings to understand the mechanisms underlying their success (Rogers et al., 2020; Clark et al., 2019).

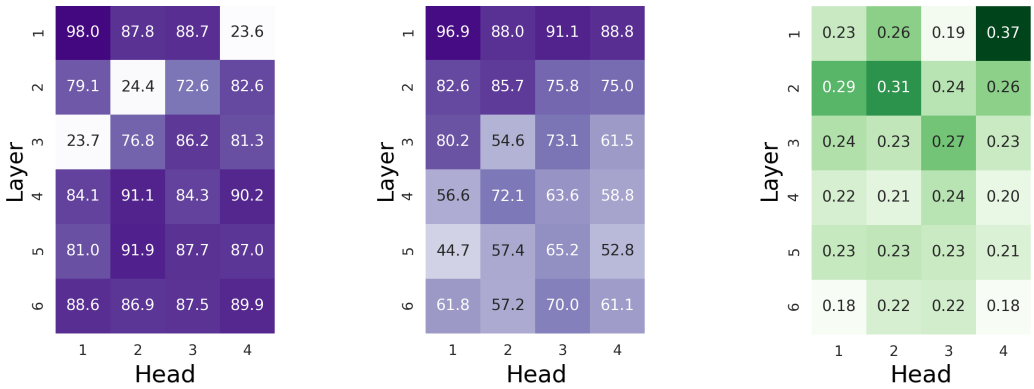
There are mainly two probing perspectives to investigate these models; they differ based on whether the target of the analysis is per-token level or it considers token-to-token interactions. The first category assesses a single word or phrase-level linguistic capabilities of BERT, such as its performance on part-of-speech tagging and word sense disambiguation performance (Tenney et al., 2019; Jawahar et al., 2019; Reif et al., 2019; Lin et al., 2019; Wallace et al., 2019).

The latter category explores the ability of Transformers to capture token-to-token interactions, such as syntactic relations and word alignment in the translation (Clark et al., 2019; Kovaleva et al., 2019; Htut et al., 2019; Reif et al., 2019; Lin et al., 2019; Goldberg, 2019; Ding et al., 2019; Zenkel et al., 2019; Li et al., 2019; Raganato and Tiedemann, 2018). The present study is closely related to the latter group; we have provided insights into the token-to-token attention in Transformer-based systems.

6.2 Analyzing the token-to-token interaction

Two types of methods are mainly considered to analyze the token-to-token interactions in Transformers. One is to track the attention weights, and the other is to check the gradient of the output with respect to the input of attention mechanisms.

Weight-based analysis: Many studies have analyzed the linguistic capabilities of Transformers by tracking attention weights. This type of analysis has covered a wide range of subjects, including syntactic and semantic relationships (Tang et al., 2018; Raganato and Tiedemann, 2018; Clark et al., 2019; Reif et al., 2019; Jawahar et al., 2019; Htut et al., 2019; Kovaleva et al., 2019; Mareček and Rosa, 2019). However, as outlined in Section 2.3, these studies have ignored the effect of $f(\boldsymbol{x})$. It has been actively discussed so far whether the attention weights can be interpreted to explain



(a) AER in the AWI setting. (b) AER in the AWO setting. (c) Averaged $\|\alpha f(\boldsymbol{x})\|$.

Figure 8: AER scores and averaged $\|\alpha f(\boldsymbol{x})\|$ in each head on one out of five seeds. The closer the extracted word alignment is to the reference, the lower the AER score—the lighter the color. The larger the averaged $\|\alpha f(\boldsymbol{x})\|$, the darker the color.

the models (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegreffe and Pinter, 2019; Pruthi et al., 2020; Vashishth et al., 2019).

Brunner et al. (2020) have introduced "effective attention," which has upgraded the weight-based analysis. Their proposal is similar to ours; they exclude attention weights that do not affect the output owing to the application of transformation f and input $\boldsymbol{x}$ in the analysis. However, our proposal differs from theirs in some aspects. Specifically, we aim to analyze the behavior of the whole attention mechanism more accurately, whereas they aim to make the attention weights more accurate. Furthermore, the effectiveness of their approach depends on the length of an input sequence; however, ours approach does not have such a limitation (see Appendix G). Additionally, we incorporate the scaling effects of f and $\boldsymbol{x}$, whereas Brunner et al. (2020) have considered only the binary effect—either the weight is canceled or not.

Gradient-based analysis: In the gradient analysis, the contribution of the input with respect to the output of the attention mechanism is calculated using the norm of a gradient matrix between the input and the output vector (Pascual et al., 2020). Intuitively, such gradient-based methods measure the change in the output vector with respect to the perturbations in the input vector. Estimating the contribution of $\boldsymbol{a}$ to $\boldsymbol{b} = \sum k\boldsymbol{a}$ by computing the gradient $\partial \boldsymbol{b}/\partial \boldsymbol{a}$ $(= k)$ is analogous to estimating the contribution of $\boldsymbol{x}$ to $\boldsymbol{y} = \sum \alpha f(\boldsymbol{x})$ by observing only an attention weight α.[17] The two approaches have the same kind of problems; that is, both ignore the magnitude of the input, $\boldsymbol{a}$ or $f(\boldsymbol{x})$.

7 Conclusions and future work

This paper showed that attention weights alone are only one of two factors that determine the output of attention. We proposed the incorporation of another factor, the transformed input vectors. Using our norm-based method, we provided a more detailed interpretation of the inner workings of Transformers, compared to the studies using the weight-based analysis. We hope that this paper will inspire researchers to have a broader view of the possible methodological choices for analyzing the behavior of Transformer-based models.

We believe that these findings can provide insights not only into the interpretation of the behaviors of Blackbox NLP systems but also into developing a more sophisticated Transformer-based system. One possible direction is to design an attention mechanism that can collect almost no information from an input sequence as the current systems achieve it by exploiting the `[SEP]` token.

In future work, we plan to apply our norm-based analysis to attention in other models, such as fine-tuned BERT, RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020). Furthermore, we expect to extend the scope of analysis from the attention to an entire Transformer architecture to better understand the inner workings and linguistic capabilities of the current powerful systems in NLP.

[17] For simplicity, we consider a linear example: $\boldsymbol{b} = \sum k\boldsymbol{a}$. We are aware that there is a gap between the two examples in terms of linearity. Further exploration of the connection to the gradient-based method is needed.

Acknowledgments

We would like to thank the anonymous reviewers of the EMNLP 2020 and the ACL 2020 Student Research Workshop (SRW), and the SRW mentor Junjie Hu for their insightful comments. We also thank the members of Tohoku NLP Laboratory for helpful comments. This work was supported by JSPS KAKENHI Grant Number JP19H04162. This work was also partially supported by a Bilateral Joint Research Program between RIKEN AIP Center and Tohoku University.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450*.

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On Identifiability in Transformers. In *8th International Conference on Learning Representations (ICLR)*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven Word Alignment Interpretation for Neural Machine Translation. In *Proceedings of the 4th Conference on Machine Translation (WMT)*, pages 1–12.

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 (NACCL-HLT)*, pages 644–648.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *arXiv preprint arXiv:1901.05287*.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do Attention Heads in BERT Track Syntactic Dependencies? *arXiv preprint arXiv:1911.12246*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3543–3556.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3651–3657.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. 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 4364–4373.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations (ICLR)*.

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 (ACL)*, pages 1293–1303.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting Inside BERT's Linguistic Knowledge. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

David Mareček and Rudolf Rosa. 2019. From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems 32 (NIPS)*, pages 14014–14024.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2020. Telling BERT's full story: from Local Attention to Global Aggregation. *arXiv preprint arXiv:2004.05916*.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. 2020. Learning to Deceive with Attention-Based Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alessandro Raganato and Jörg Tiedemann. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. *Advances in Neural Information Processing Systems 32 (NIPS)*, pages 8594–8603.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*.

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 (ACL)*, pages 1715–1725.

Sofia Serrano and Noah A Smith. 2019. Is Attention Interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2931–2951.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An Analysis of Attention Mechanisms: The Case of Word Sense Disambiguation in Neural Machine Translation. In *Proceedings of the 3rd Conference on Machine Translation (WMT): Research Papers*, pages 26–35.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations (ICLR)*.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention Interpretability Across NLP Tasks. *arXiv preprint arXiv:1909.11218*.

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 (NIPS)*, pages 5998–6008.

David Vilar, Maja Popović, and Hermann Ney. 2006. AER: Do we need to "improve" our alignments? In *International Workshop on Spoken Language Translation (IWSLT) 2006*, pages 205–212.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. 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 5307–5315.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. 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 11–20.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32 (NIPS)*, pages 1–18.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. *arXiv preprint arXiv:1901.11359*.

A Multi-head attention and the norm-based analysis

Our norm-based analysis is applicable to the analysis of the multi-head attention mechanism implemented in Transformers. The i-th output of the multi-head attention mechanism $\boldsymbol{y}_i^{\text{integrated}}$ is calculated as follows:

$$\boldsymbol{y}_i^{\text{integrated}} = \sum_h \boldsymbol{y}_i^h \tag{7}$$

$$\boldsymbol{y}_i^h = \sum_{j=1}^{n} \alpha_{i,j}^h f^h(\boldsymbol{x}_j) \tag{8}$$

$$f^h(\boldsymbol{x}) := \left(\boldsymbol{x}\boldsymbol{W}^{V,h} + \boldsymbol{b}^{V,h}\right)\boldsymbol{W}^{O,h}, \tag{9}$$

where $\alpha_{i,j}^h$, $\boldsymbol{W}^{V,h}$, $\boldsymbol{b}^{V,h}$, and $\boldsymbol{W}^{O,h}$ are the same as $\alpha_{i,j}$, $\boldsymbol{W}^V$, $\boldsymbol{b}^V$, and $\boldsymbol{W}^O$ in Equations 3 and 4 for each head h, respectively. n is the number of tokens in the input vectors. Equation 7 can be rewritten as follows:

$$\boldsymbol{y}_i^{\text{integrated}} = \sum_{j=1}^{n} \sum_h \alpha_{i,j}^h f^h(\boldsymbol{x}_j) \tag{10}$$

As shown in Equation 10, the multi-head attention mechanism is also linearly decomposable, and one can analyze the flaw of the information from the j-th vector to the i-th vector by measuring $\|\sum_h \alpha_{i,j}^h f^h(\boldsymbol{x}_j)\|$. In Section 5, we actually used $\|\sum_h \alpha_{i,j}^h f^h(\boldsymbol{x}_j)\|$ to extract the alignment from each layer's multi-head attention.

The output of the multi-head attention mechanism is calculated via the sum of the outputs of all the heads and a bias $\boldsymbol{b}^O \in \mathbb{R}^d$. Because adding a fixed vector is irrelevant to the token-to-token interaction that we aim to investigate, we omitted $\boldsymbol{b}^O$ in our analysis.

B The source of the dispersion of $\|f(\boldsymbol{x})\|$

As described in Section 4.1, $\|f(\boldsymbol{x})\|$ exhibits dispersion; however, it remains unclear whether this dispersion is attributed to $\|\boldsymbol{x}\|$ or f. Hence, we checked the dispersion of $\|\boldsymbol{x}\|$ and the scaling effects of the transformation f.

Dispersion of $\|\boldsymbol{x}\|$: First, we checked the coefficient of variation (CV) of $\|\boldsymbol{x}\|$. Table 4 shows that the average CV is 0.12, which is less than that of $\|f(\boldsymbol{x})\|$ (0.22). The value of $\|\boldsymbol{x}\|$ typically varies between 0.88 and 1.12 times the average value of $\|\boldsymbol{x}\|$. The layer normalization (Ba et al., 2016) that applied at the end of the previous layer should have a large impact on the variance of $\|\boldsymbol{x}\|$.

Scaling effects of f: Second, we investigated the scaling effect of the transformation f on the norm of the input. Because the affine transformation $f\colon \mathbb{R}^d \to \mathbb{R}^d$ can be considered a linear transformation $\mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$ (Appendix C), the singular values of f can be regarded as its scaling effect. Figure 9 shows the singular values of f in randomly selected heads in BERT. The singular values are displayed in descending order from left to right. In each head, there is a difference of at least 1.8 times between the maximum and minimum singular values. This difference is larger than that of $\|\boldsymbol{x}\|$, where $\|\boldsymbol{x}\|$ typically varies between 0.88 and 1.12 times the average value. These results suggest that the dispersion of $\|f(\boldsymbol{x})\|$ is primarily attributed to the scaling effect of f.

C Affine transformation as linear transformation

The affine transformation $f\colon \mathbb{R}^d \to \mathbb{R}^d$ in Equation 4 can be viewed as a linear transformation $\widetilde{f}\colon \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$. Given $\widetilde{\boldsymbol{x}} := \begin{bmatrix} \boldsymbol{x} & 1 \end{bmatrix} \in \mathbb{R}^{d+1}$, where 1 is concatenated to the end of each input vector $\boldsymbol{x} \in \mathbb{R}^d$, the affine transformation f can be viewed as:

$$\widetilde{f}(\widetilde{\boldsymbol{x}}) = \widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{W}}^V\widetilde{\boldsymbol{W}}^O \tag{11}$$

$$\widetilde{\boldsymbol{W}}^V := \begin{bmatrix} & & 0 \\ & \boldsymbol{W}^V & \vdots \\ & & 0 \\ & \boldsymbol{b}^V & 1 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d'+1)} \tag{12}$$

$$\widetilde{\boldsymbol{W}}^O := \begin{bmatrix} & & 0 \\ & \boldsymbol{W}^O & \vdots \\ & & 0 \\ 0 & \dots & 0 \quad 1 \end{bmatrix} \in \mathbb{R}^{(d'+1)\times(d+1)}. \tag{13}$$

D Details on Sections 4.2 and 4.3

We describe the detailed experimental setup presented in Sections 4.2 and 4.3.

D.1 Notations

The dataset consists of several sequences; Data $= (s_1, \cdots, s_{|Data|})$. Each sequence consists of sev-

Layer	μ	σ	CV	Max	Min
12 (max CV)	20.49	4.62	**0.23**	32.84	4.13
7 (min CV)	21.64	1.40	**0.06**	23.03	11.87
Average	19.93	2.39	**0.12**	-	-

Table 4: Mean (μ), standard deviation (σ), coefficient of variance (CV), and maximum and minimum values of $\|\mathbf{x}\|$; the former three are averaged on all the layers.

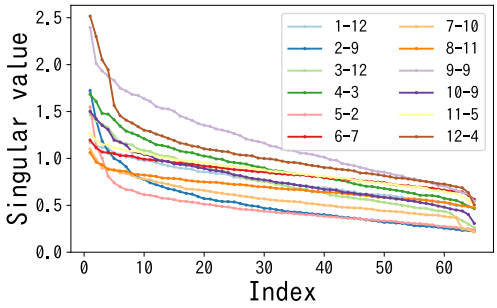


Figure 9: Singular values of f at randomly selected heads in each layer. We use ⟨layer⟩-⟨head number⟩ to denote a particular attention head. The singular values are

eral tokens, $s_p = (t_1^p, \cdots, t_{|s_p|}^p)$, where t_q^p is the q-th token in the p-th sequence. For simplicity, we define the following functions:

$$\text{Weight}(p,q,\ell,h) = \frac{1}{|s_p|}\sum_{i=1}^{|s_p|}\alpha_{p,i,q}^{\ell,h}$$
$$\text{Norm}(p,q,\ell,h) = \|f^{\ell,h}(\boldsymbol{x}_{p,q}^{\ell})\|$$
$$\text{WNorm}(p,q,\ell,h) = \frac{1}{|s_p|}\sum_{i=1}^{|s_p|}\|\alpha_{p,i,q}^{\ell,h} f^{\ell,h}(\boldsymbol{x}_{p,q}^{\ell})\|,$$

where $\alpha_{p,i,q}^{\ell,h}$ is the attention weight assigned from the i-th pre-update vector to the q-th input vector in the p-th sequence. h and ℓ denote that the score is obtained from the h-th head of the ℓ-th layer. $\boldsymbol{x}_{p,q}^{\ell}$ denotes the input vector for token t_q^p in the ℓ-th layer. $f^{\ell,h}(\boldsymbol{x}_{p,q}^{\ell})$ is the transformed vector for $\boldsymbol{x}_{p,q}^{\ell}$ in the h-th head of the ℓ-th layer.

Next, the vocabulary $\mathcal{V}$ of BERT is divided into the following four categories:

$$A = \{\texttt{[CLS]}\}$$
$$B = \{\texttt{[SEP]}\}$$
$$C = \{\text{“.”}, \text{“,”}\}$$
$$D = \mathcal{V} \setminus (A \cup B \cup C). \quad (14)$$

Let $T(p, Z)$ be a function that returns all tokens t_q^p belonging to the category Z in the p-th sequence. To formally describe our experiments, several functions are defined as follows. Note that we analyzed a model with 12 heads in each layer.

$$\text{MeanN}(Z,\ell,h,p) = \frac{1}{|T(Z,p)|}\sum_{t_q^p \in T(Z,p)} \text{Norm}(p,q,\ell,h)$$
$$\text{SumW}(Z,\ell,h,p) = \sum_{t_q^p \in T(Z,p)} \text{Weight}(p,q,\ell,h)$$
$$\text{SumWN}(Z,\ell,h,p) = \sum_{t_q^p \in T(Z,p)} \text{WNorm}(p,q,\ell,h)$$
$$\text{HeadN}(Z,\ell,h) = \frac{1}{|\text{Data}|}\sum_{s_p \in \text{Data}} \text{MeanN}(Z,\ell,h,p)$$
$$\text{HeadW}(Z,\ell,h) = \frac{1}{|\text{Data}|}\sum_{s_p \in \text{Data}} \text{SumW}(Z,\ell,h,p)$$
$$\text{HeadWN}(Z,\ell,h) = \frac{1}{|\text{Data}|}\sum_{s_p \in \text{Data}} \text{SumWN}(Z,\ell,h,p)$$
$$\text{LayerW}(Z,\ell) = \frac{1}{12}\sum_{h=1}^{12} \text{HeadW}(Z,\ell,h)$$
$$\text{LayerWN}(Z,\ell) = \frac{1}{12}\sum_{h=1}^{12} \text{HeadWN}(Z,\ell,h).$$

The LayerW$(\cdot)$ and LayerWN$(\cdot)$ functions are used to analyze the average behavior of the heads in a layer.

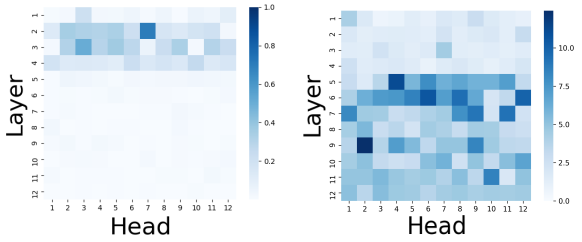
D.2 Experimental setup for Section 4.2

In Figure 3, the results of each layer are reported for each category. In Figures 3a and 3b, the values for each category Z were calculated using LayerW(Z,ℓ) and LayerWN(Z,ℓ), respectively.

In Figure 4, α and $\|f(\boldsymbol{x})\|$ in the h-th head of the ℓ-th layer were calculated using HeadW(Z,ℓ,h) and HeadN(Z,ℓ,h), respectively. The scores reported in Table 2 are the Spearman rank correlation coefficient r between Weight(p,q,ℓ,h) and WNorm(p,q,ℓ,h). We calculated the r using all the pairs of Weight(p,q,ℓ,h) and WNorm(p,q,ℓ,h) for the possible values of p, q, ℓ, and h. In Figure 5, each plot corresponds to the pair of Weight(p,q,ℓ,h) and WNorm(p,q,ℓ,h), where the combination of (p,q,ℓ,h) was randomly determined.

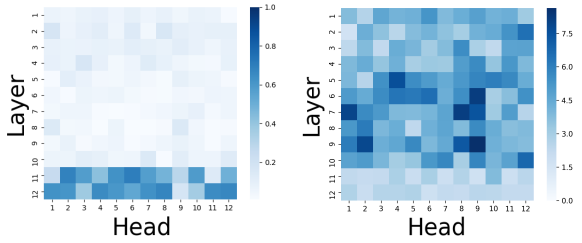
D.3 Visualizations of α and $\|f(\boldsymbol{x})\|$ for each word category

As described in Section 4.2, α and $\|f(\boldsymbol{x})\|$ for the [SEP] token were canceled out in almost all heads (Figure 4). Here, we show the trends for the other categories—B, C, and D in Equation 14. Figures 10, 11, and 12 show the trends of α and $\|f(\boldsymbol{x})\|$ for category B (the [CLS] token), C (periods and commas), and D (other tokens), respec-



(a) α. (b) $\|f(\boldsymbol{x})\|$.

Figure 10: α and $\|f(\boldsymbol{x})\|$ corresponding to `[CLS]` token, averaged on all the input text.



(a) α. (b) $\|f(\boldsymbol{x})\|$.

Figure 11: α and $\|f(\boldsymbol{x})\|$ corresponding to periods and commas, averaged on all the input text.

tively. The values in these figures were calculated as described in Appendix D.2. Figures 10 and 11 show that the trends for categories B and C were analogous to those for the `[SEP]` token; the large α was canceled by the small $\|f(\boldsymbol{x})\|$. However, the trends for category D do not exhibit the trends of the negative correlation between α and $\|f(\boldsymbol{x})\|$. In each heatmap of $\|f(\boldsymbol{x})\|$, the color scale is determined by the maximum value of $\|f(\boldsymbol{x})\|$ in each category.

We also reported the relationship between α and $\|f(\boldsymbol{x})\|$ in Section 4.2 (Figure 5). Figure 13 shows the results for each word category to provide a clearer display of the results.

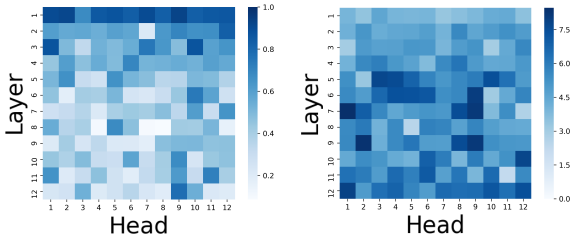
D.4 Experimental setup and visualizations for Section 4.3

In Section 4.3, we analyzed the relationship between the word frequency and $\|f(\boldsymbol{x})\|$. To formally describe our experiments, we further define the functions as follows:

$$\mathrm{AvgW}(p,q) = \frac{1}{12\cdot 12}\sum_{\ell=1}^{12}\sum_{h=1}^{12}\mathrm{Weight}(p,q,\ell,h)$$

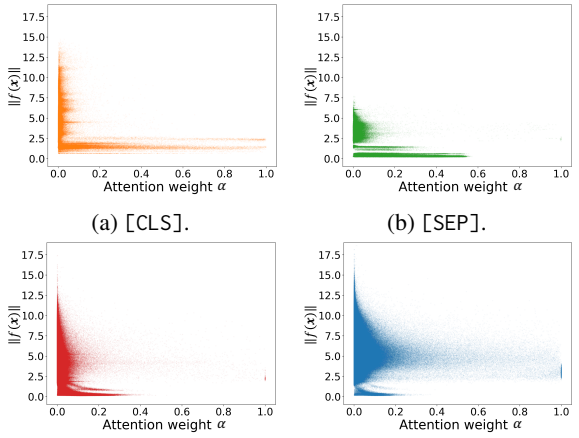
$$\mathrm{AvgN}(p,q) = \frac{1}{12\cdot 12}\sum_{\ell=1}^{12}\sum_{h=1}^{12}\mathrm{Norm}(p,q,\ell,h).$$

Note that we analyzed a model comprising 12 layers; each layer has 12 attention heads. Let



(a) α. (b) $\|f(\boldsymbol{x})\|$.

Figure 12: α and $\|f(\boldsymbol{x})\|$ corresponding to other tokens, averaged on all the input text.



(a) `[CLS]`. (b) `[SEP]`.

(c) Periods and commas. (d) Other tokens.

Figure 13: Relationship between α and $\|f(\boldsymbol{x})\|$ for each category.

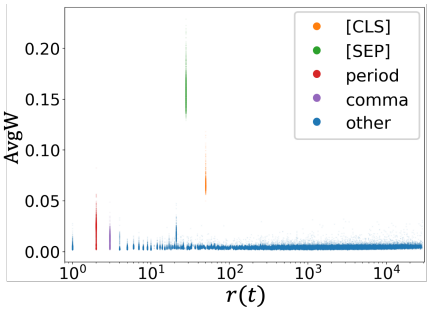
$r(\cdot)$ be a function that returns the frequency rank of a given word. We first calculated the Spearman rank correlation coefficient between $r(t_q^p)$ and $\mathrm{AvgW}(p,q)$. The score was 0.06, which suggests that there is no relationship between α and the frequency rank of the word. Then, we calculated the Spearman rank correlation coefficient between $r(t_q^p)$ and $\mathrm{AvgN}(p,q)$. The score was 0.75, which suggests a strong correlation between $\|f(\boldsymbol{x})\|$ and the frequency rank of the word; Figure 14 shows these results.

In addition, the results for the word frequency, instead of the frequency rank, are shown in Figure 15. $c(\cdot)$ denotes a function that returns the frequency of a given word in the training dataset of BERT. We reproduced the dataset because it is not released.

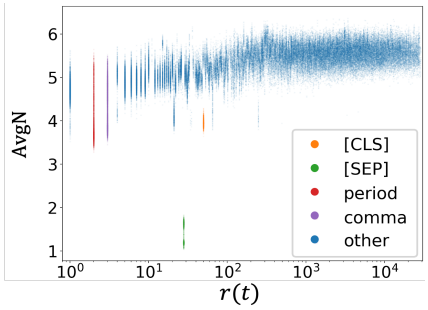
E Details on Section 5

E.1 Hyperparameters and training settings

We used the Transformer (Vaswani et al., 2017) NMT model implemented in fairseq (Ott et al., 2019) for the experiments. Table 5 shows the hyperparameters of the model, which were the same

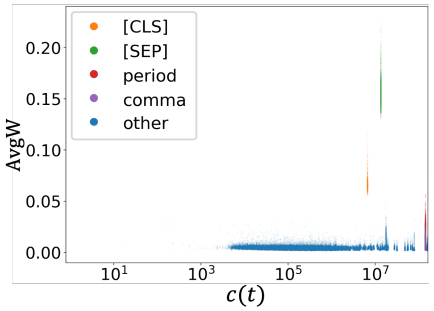


(a) Relationship between $r(t)$ and AvgW.

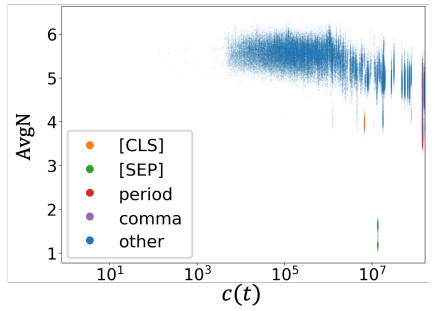


(b) Relationship between $r(t)$ and AvgN.

Figure 14: Relationship between frequency rank $r(t_q^p)$ and AvgW(p, q), and that between $r(t_q^p)$ and AvgN(p, q).



(a) Relationship between $c(t)$ and AvgW.



(b) Relationship between $c(t)$ and AvgN.

Figure 15: Relationship between frequency count $c(t_q^p)$ and AvgW(p, q), and that between $c(t_q^p)$ and AvgN(p, q).

as those used by Ding et al. (2019). We used the model with the highest BLEU score in the development set for our experiments.

We conducted the data preprocessing[18] following the method by Zenkel et al. (2019) and Ding et al. (2019). All the words in the training data of the NMT systems were split into subword units using byte-pair encoding (BPE, Sennrich et al. (2016)) with 10k merge operations. Following Ding et al. (2019), the last 1000 instances of the training data were used as the development data.

E.2 Settings of the word alignment extraction

First, we applied BPE, which was used to split the training data of the NMT systems to create the evaluation data used for calculating the AER scores. Next, we extracted the scores of α and $\|\alpha f(\boldsymbol{x})\|$ for each subword in the evaluation data for the force decoding setup. The gold alignments are annotated at the word-level, not the subword-level. To calculate the word-level alignment scores, α and $\|\alpha f(\boldsymbol{x})\|$ for the subwords were merged along with the target token in the gold data by averaging, then merged along with the source tokens in the gold data by summation. These operations were the same as Li et al. (2019).

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

In existing studies, ⟨/s⟩, the special token for the end of the sentence, was probably removed in calculating word alignments. We included ⟨/s⟩ as the alignment targets and we considered the alignments to ⟨/s⟩ as no alignment. In other words, if the model aligns a certain word with ⟨/s⟩, we assume that the model decides that the word is not aligned to any word.

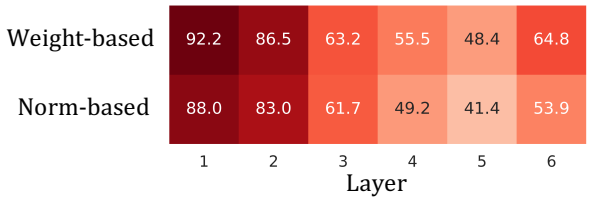
E.3 Layer-wise analysis

We preliminarily investigated how the source-target attentions in a Transformer-based NMT system behave depending on the layer. Tang et al. (2018) have reported that they behave differently depending on the layer. The AER scores in the AWI and AWO settings were calculated for each layer (Figure 16). In the AWO setting, AER scores tend to be better in the latter layers than in the earlier layers (Figure 16a). In contrast, the AER scores tend to be better in the earlier layers than in the latter layers in the AWI setting (Figure 16b).

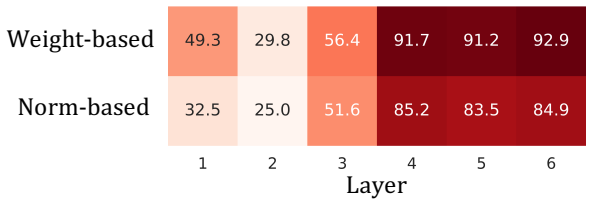
These results suggest that the earlier and latter layers focus on the source word that is aligned with the input and output target word, respectively (as shown in Figure 6). Furthermore, we believe that it is a convincing result to extract cleaner word alignments from the AWI setting than the AWO setting (Figure 16), because the AWI setting is

Fairseq model	architecture	transformer_iwslt_de_en
	encoder embed dim.	512
	decoder embed dim.	512
	encoder ffn embed dim.	1024
	decoder ffn embed dim.	1024
	encoder attention heads	4
	decoder attention heads	4
	encoder layers	6
	decoder layers	6
Activation	function	Relu
Loss	type	label smoothed cross entropy
	label smoothing	0.1
Optimizer	algorithm	Adam
	learning rates	0.001
	β_1	0.9
	β_2	0.98
	weight decay	0.0
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	4,000
	warmup init lrarning rate	1e-07
Training	batch size	80
	max tokens	4000
	max epoch	100
	update freq	8
	drop out	0.1
	seed	2
	number of GPUs used	2

Table 5: Hyperparameters of the NMT model.



(a) AWO setting.



(b) AWI setting.

Figure 16: Layer-wise AER scores. Each value is the average of five random seeds. The closer the extracted word alignment is to the reference, the lower the AER score—the lighter the color.

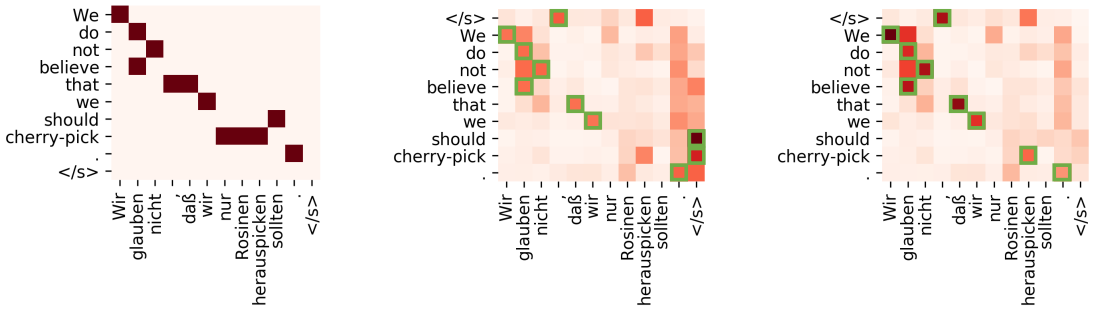
more advantageous. The main advantage is that while the decoder may fail to predict the correct output words, the input words are perfectly accurate owing to the teacher forcing.

E.4 Alignments in different layers

Figures 17 to 22 show additional examples of the extracted alignments from the different layers of the NMT system. Note that the color scale in each heatmap is determined by the maximum value in each figure. One can observe that while the attention weights α are biased towards $\langle$/s$\rangle$, the norms $\|\alpha f(\boldsymbol{x})\|$ corresponding to the token are small.

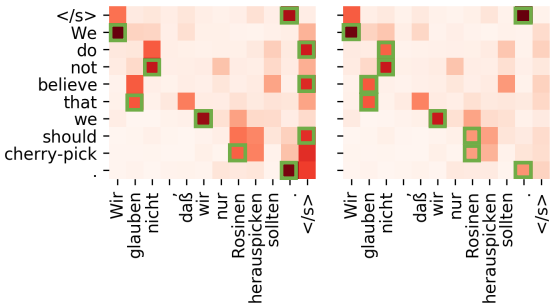
F Word alignment experiments on different settings

To verify whether the results obtained in the Section 5 are reproducible in different settings, we conducted an additional experiment using the model with a different number of attention heads. Specifically, we used a model with eight attention heads in both the encoder and decoder. Table 6 shows the AER scores of the 8-head model. As with the results obtained by the 4-head model, word alignments extracted using the proposed norm-based approach were more reasonable than those extracted using the weight-based approach, and better word alignments are extracted in the AWI setting than in the AWO setting. Furthermore, the alignments extracted using the head or the layer with the highest average $\|\alpha f(\boldsymbol{x})\|$ in the AWI setting are competitive with one of the existing word aligners—fast_align. With respect to the weight-based extraction, the scores obtained using



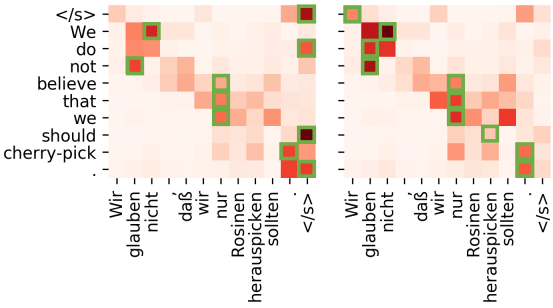
(a) Reference. (b) Attention-weights. (c) Vector-norms (ours).

Figure 17: Examples of the reference alignment and the extracted patterns by each method in layer 1. Word pairs with a green frame shows the word with the highest weight or norm. The vertical axis represents the input source word in the decoder, and the pairs with a green frame are extracted as alignments in the AWI setting. Note that pairs that contain ⟨/s⟩ not extracted.



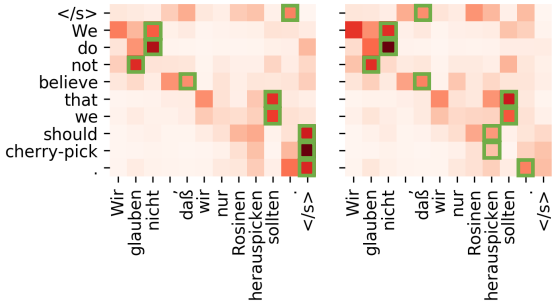
(a) Attention-weights. (b) Vector-norms.

Figure 18: Examples of the reference alignment and the extracted patterns by each method in layer 2.



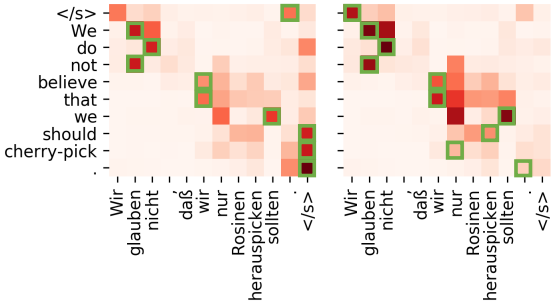
(a) Attention-weights. (b) Vector-norms.

Figure 20: Examples of the reference alignment and the extracted patterns by each method in layer 4.



(a) Attention-weights. (b) Vector-norms.

Figure 19: Examples of the reference alignment and the extracted patterns by each method in layer 3.



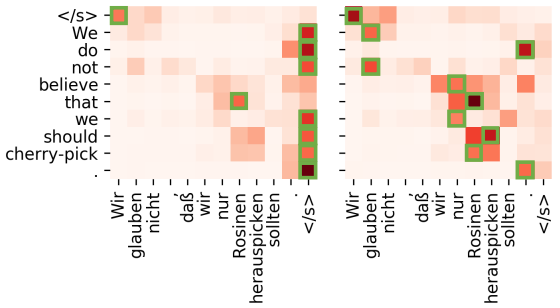
(a) Attention-weights. (b) Vector-norms.

Figure 21: Examples of the reference alignment and the extracted patterns by each method in layer 5.

the 8-head model were worse than those obtained using the 4-head model. This may be owing to the increase in the number of heads that do not capture reasonable alignments.

Figures 23a and 23b show the AER scores of the alignments obtained by the norm-based extraction at each head on one out of five seeds. Figure 23c shows the average of $\|\alpha f(\boldsymbol{x})\|$ at each head. As with the results obtained by the 4-head model, the heads with the low (i.e., better) AER score in the AWI setting tended to have the high $\|\alpha f(\boldsymbol{x})\|$ (the Spearman rank and Pearson correlation coefficients between the AER scores and averaged $\|\alpha f(\boldsymbol{x})\|$ among the 6×8 heads are -0.26 and -0.50). In contrast, in the AWO setting, such a negative correlation is not observed; rather, a positive correlation is observed (the Spearman's ρ is 0.40 and the Pearson's r is 0.40).

Additionally, following Appendix E.3, the AER scores for both the AWI and AWO settings for each layer were calculated (Figure 24). As with the 4-head model (Appendix E.3), the latter layers correspond to the AWO setting and the earlier layers corresponds to the AWI setting in the 8-head

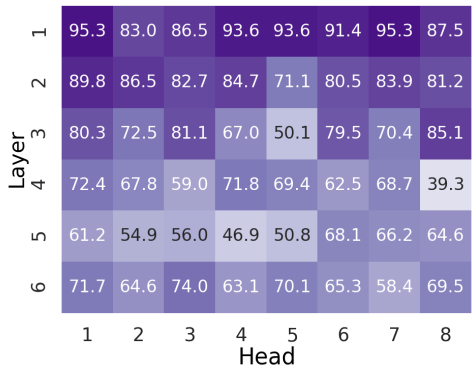


(a) Attention-weights. (b) Vector-norms.

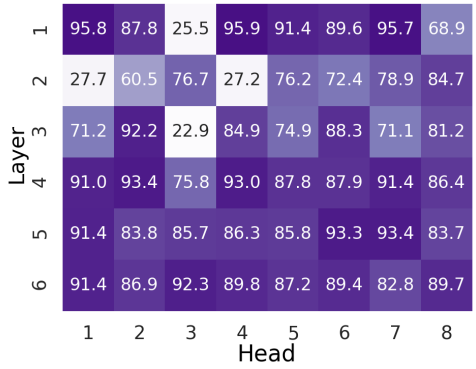
Figure 22: Examples of the reference alignment and the extracted patterns by each method in layer 6.

Methods	AER	±SD
Transformer – Attention-based Approach		
— *Alignment with output* setting —		
Weight-based		
layer mean	70.4	0.6
best layer (layer 4 or 5)	49.3	1.2
Norm-based (ours)		
layer mean	63.2	0,7
best layer (layer 5)	43.4	0.8
head with the highest average $\|\alpha f(\boldsymbol{x})\|$	87.2	0.6
layer with the highest average $\|\alpha f(\boldsymbol{x})\|$	83.7	2.2
— *Alignment with input* setting —		
Weight-based		
layer mean	76.6	1.7
best layer (layer 2 or 3)	38.7	8.9
Norm-based (ours)		
layer mean	59.9	1.0
best layer (layer 2 or 3)	26.3	1.9
head with the highest average $\|\alpha f(\boldsymbol{x})\|$	24.9	1.7
layer with the highest average $\|\alpha f(\boldsymbol{x})\|$	26.5	1.9
Word Aligner		
fast_align from Zenkel et al. (2019)	28.4	-
GIZA++ from Zenkel et al. (2019)	21.0	-

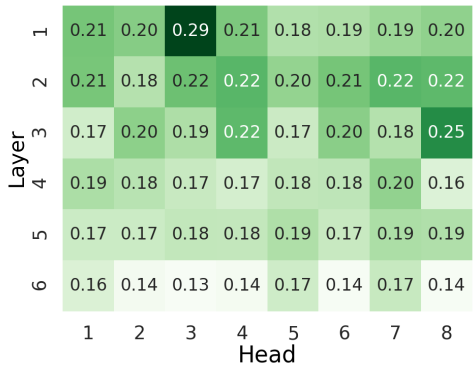
Table 6: Results on a model trained with the same settings as described in Appendix E.1 except that the number of attention heads in the encoder and decoder is 8. Each value is the average of five random seeds.



(a) AER in the AWO setting.



(b) AER in the AWI setting.



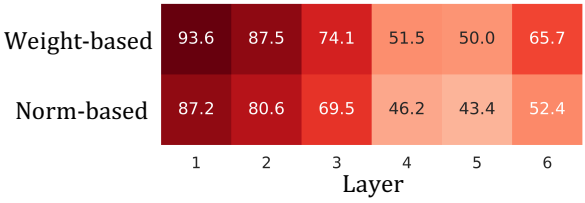
(c) Averaged $\|\alpha f(\boldsymbol{x})\|$.

Figure 23: AER scores and averaged $\|\alpha f(\boldsymbol{x})\|$ for each head in a model with 8 heads.

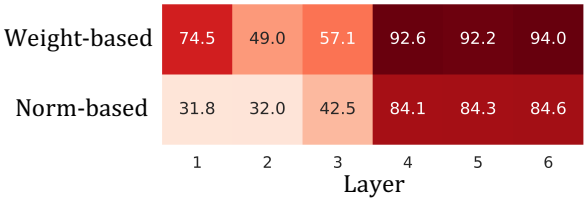
model.

G Comparison with effective attention (Brunner et al., 2020)

In this section, we discuss the difference between our approach and "effective attention" (Brunner et al., 2020), which is an enhanced version of the weight-based analysis. The effective attention exclude the components that do not affect the output owing to the application of transformation f and input $\boldsymbol{x}$ from the attention weight matrix $\boldsymbol{A}$. The output-irrelevant components are derived from the null space of the matrix $\boldsymbol{T}$, which is the stack of $f(\boldsymbol{x})$. Figure 25a shows the Pearson correlation coefficient between the raw attention weight and the effective attention. Since the dimension of the null space of the matrix $\boldsymbol{T}$ depends on the length of the input sequence, as shown in Figure 25a, the effective attention and raw attention weight are identical for short input sequences. Figure 25b shows the Pearson correlation coefficient between the raw attention weight and our norm-based method. Since we incorporate the scaling effects of f and $\boldsymbol{x}$, which contain canceling, our proposed method $\|\alpha f(\boldsymbol{x})\|$ differs from the raw attention weight, whether the input sequence is long or short.

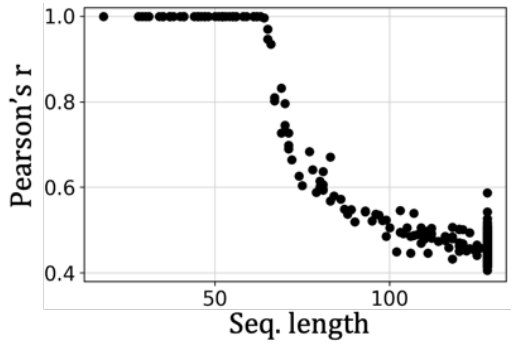


(a) AWO setting.

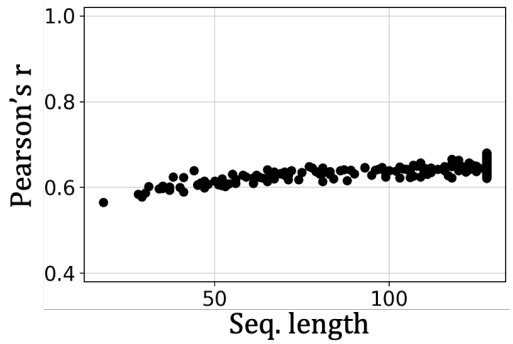


(b) AWI setting.

Figure 24: Layer-wise AER scores. Each value is the average of five random seeds. The closer the extracted word alignment is to the reference, the lower the AER score—the lighter the color.



(a) Effective attention.



(b) $\|\alpha f(\boldsymbol{x})\|$.

Figure 25: Each point represents the Pearson correlation coefficient of raw attention and each method toward token length.