Attention Analysis and Calibration for Transformer in Natural Language Generation

Yu Lu, Jiajun Zhang, *Senior Member, IEEE*, Jiali Zeng, Shuangzhi Wu, and Chengqing Zong, *Senior Member, IEEE*

***Abstract*— Attention mechanism has been ubiquitous in neural machine translation by dynamically selecting relevant contexts for different translations. Apart from performance gains, attention weights assigned to input tokens are often utilized to explain that high-attention tokens contribute more to the prediction. However, many works question whether this assumption holds in text classification by manually manipulating attention weights and observing decision flips. This article extends this question to Transformer-based neural machine translation, which heavily relies on cross-lingual attention to produce accurate translations but is relatively understudied in this context. We first design a mask perturbation model which automatically assesses each input's contribution to model outputs. We then test whether the token contributing most to the current translation receives the highest attention weight. We find that it sometimes does not, which closely depends on the entropy of attention weights, the syntactic role of the current generation, and language pairs. We also rethink the discrepancy between attention weights and word alignments from the view of unreliable attention weights. Our observations further motivate us to calibrate the cross-lingual multi-head attention by attaching more attention to indispensable tokens, whose removal leads to a dramatic performance drop. Empirical experiments on different-scale translation tasks and text summarization tasks demonstrate that our calibration methods significantly outperform strong baselines.**

***Index Terms*—Attention mechanism, interpretability, Transformer, attention calibration.**

I. Introduction

Attention mechanism [1] has become a ubiquitous component of natural language processing (NLP) tasks, especially for neural machine translation (NMT). It computes conditional distributions over inputs to obtain a weighted context vector for downstream modules. In addition to performance improvements, attention weights are often implicitly or explicitly claimed to explain the model's decision-making process: inputs assigned with large attention weights contribute more to the output. Such claims that attention provides explanation are common in the literature [2]–[4].

However, many recent studies challenge whether attention can be an explanation. The underlying question seems to be: *do high-attention weights on specific inputs lead the model to make its prediction?* Some studies observe decision flips by manually perturbing attention weights and claim that the answer is surprisingly no [5], [6]. Another opposite idea is that trained attention weights do learn something meaningful about relations between inputs and outputs [7]. Existing discussions are mainly based on text classification tasks, with a focus on classical attention functions. In this paper, we extend this question to Transformer-based NMT [8], which heavily relies on cross-lingual attention weights to generate correct translations. As a core element of Transformer, multi-head attention, a new attention variant, may have different interpretability properties, which are not systematically explored.

Attention Analysis. To answer the above question in Transformer, we plan to identify the most informative inputs and compare them with high-attention tokens. Specifically, we propose to observe how the model decision changes as perturbing parts of inputs. We define the perturbation operation as applying a learnable mask to scale each attention weight. Then, we perform a "deletion game" to find the smallest perturbation extents that cause significant quality degradation. In this manner, we can find the most informative inputs for the output and then compare them with high-attention tokens. If they are the same, we can treat attention as a reliable explanation and vice versa.

We take Fig. 1 as an example. After producing the target word "in," the source word "远郊 [countryside]" receives a high attention weight. Our mask perturbation model finds that perturbing that word can indeed largely hamper the next generation. In this case, attention can be a reliable explanation. However, after the prediction "deaths," attention mechanisms attach most attention to "⟨EOS⟩," while source words "交通 [traffic]" and "中断 [interruption]" are truly important for the subsequent translation discovered by our mask perturbation model. Thus attention is not a reliable indicator of inputs' contributions at this timestep.

We thoroughly examine whether attention weights in Transformer can precisely indicate inputs' contributions on three benchmarks: Zh ⇒ En, En ⇒ De, and En ⇒ Fr. We find that it is not a simple "yes-or-no" question but related to three factors:

Manuscript received December 3, 2021; revised April 23, 2022; accepted May 21, 2022. Date of publication June 9, 2022; date of current version June 15, 2022. This work was supported by the Natural Science Foundation of China under Grants 62122088, U1836221, and 62006224. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfeng Gao. *(Corresponding author: Jiajun Zhang.)*

Yu Lu, Jiajun Zhang, and Chengqing Zong are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing 100190, China, and also with School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yu.lu@nlpr.ia.ac.cn; jjzhang@nlpr.ia.ac.cn; cqzong@nlpr.ia.ac.cn).

Jiali Zeng and Shuangzhi Wu are with the Department Tencent Cloud Xiaowei, Beijing 100089, China (e-mail: lemonzeng@tencent.com; frostwu@tencent.com).

Digital Object Identifier 10.1109/TASLP.2022.3180678

2329-9290 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

Src: 远郊 连 日 大雪 多 人 死亡 交通 中断
Ref: days of heavy snow in countryside left many deaths and transportation disrupted
Base: heavy snow in countryside caused many deaths
Ours: heavy snow *in* countryside has caused many *deaths* and traffic interruption

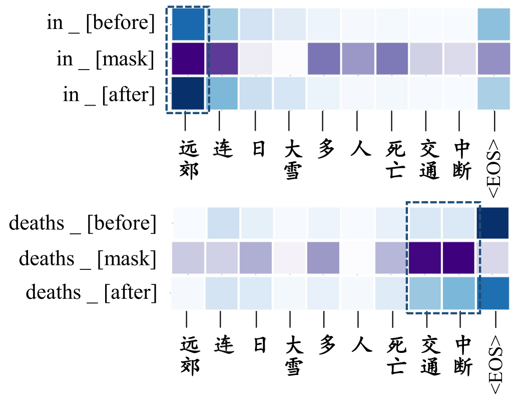


Fig. 1. Examples of attention weights before and after calibration. "in _" denotes the timestep after the prediction "in". The dashed boxes indicate inputs which are expected to receive more attention measured by our mask perturbation model.

- *The entropy of attention weights*. When the entropy is relatively higher, the likelihood that high-attention input tokens decide the model output decreases. But the declining trend differs among the three language pairs.
- *The syntactic role of the current translation*. Attention weights are more likely to indicate inputs' contributions when generating content words (e.g., nouns and verbs) than function words (e.g., conjunctions and prepositions).
- *The type of language pairs*. The chance that attention weights can be a reliable explanation is much higher in language pairs with a smaller syntactic gap.

We further explore another role of attention weights as a word alignment learned by black-box NMT models [9], [10]. Former researches show that learned attention weights diverge from word alignment, a correspondence between a pair of source and target words [11], [12]. For instance, semantically unrelated words like "⟨EOS⟩" frequently draw a wide range of attention (as seen in the bottom case in Fig. 1). Here we question whether unreliable attention weights cause this divergence. Analytical results show that alignments extracted from inputs' contributions are closer to word alignment than those from attention weights. It indicates that the model behaviour does not greatly deviate from our intuition but is misled by unreliable attention weights.

Attention Calibration: According to our observations, we find that the NMT model is prone to assign high attention weights to those tokens with limited effect on the prediction, leading to wrong-translation or over-translation in NMT [13]. Thus, we propose to calibrate the vanilla attention mechanism by focusing more on critical inputs. As mentioned earlier, the mask perturbation model helps to detect informative inputs for each prediction. Based on this, we further calibrate attention weights by reallocating more attention to informative inputs. The mask perturbation model and NMT model are jointly trained, while attention weights in NMT are corrected based on actual contributions measured by the mask perturbation model.

Recall the example in Fig. 1. For the top situation, we strengthen the attention weight of "远郊 [countryside]". But in the bottom case, we redistribute attention weights to source words ("交通 [traffic]" and "中断 [interruption]") which receive little attention but are critical for the following translation. After calibration, the missing source information "traffic interruption" is well-translated.

We verify the effectiveness of our method on extensive translation tasks (NIST Zh ⇒ En, WMT14 En ⇒ De, WMT17 En ⇔ Fi, WMT17 En ⇔ Lv, and WMT16 En ⇔ Ro) and abstractive text summarization task. Experimental results confirm that our attention calibration method achieves significant improvements over strong baselines. We further visualize calibrated attention weights and investigate which attention weights need to be corrected across different layers.

Attention calibration methods have been presented in our previous paper [14]. In this article, we make the following significant extensions to our previous work.

- We explore the capability of our mask perturbation model as an analytical tool to comprehensively study the interpretability of multi-head attention in Transformer-based NMT. We give detailed analyses to show when attention weights are reliable indicators of inputs' contributions, which provides value to the study of the model's inner working through attention mechanisms.
- We further evaluate our methods on abstractive text summarization, where attention weights are heavily relied on to search salient inputs. Our attention calibration method shows substantial advantages in this case, showing the effectiveness of our approach in different attention-based networks. It also reinforces the necessity of attention analyses and calibration in more attention-dependent tasks.

II. Background

The Transformer has a typical encoder-decoder framework with stacking layers of attention blocks. The encoder first transforms an input $x = \{x_1, x_2, \ldots x_n\}$ to a sequence of continuous representations $\boldsymbol{h} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{\ldots h_n}\}$, from which the decoder generates an output sequence $y = \{y_1, y_2, \ldots y_m\}$. Multi-head attention (MHA) between encoder and decoder enables each prediction to attend overall inputs from different representation subspaces jointly. For the single head, we first project $\boldsymbol{h} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{\ldots h_n}\}$ to keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ using different linear projections. At the t-th position, we project the hidden state of the previous decoder layer to the query vector $\boldsymbol{q}_t$. Then we multiply $\boldsymbol{q}_t$ by keys $\boldsymbol{K}$ to obtain the attention distribution $\boldsymbol{a}_t$, which is used to calculate a weighted sum of values $\boldsymbol{V}$.

$$\begin{aligned} \mathrm{Attn}\left(\boldsymbol{q}_t, \boldsymbol{K}, \boldsymbol{V}\right) &= \boldsymbol{a_t} * \boldsymbol{V} \\ \boldsymbol{a_t} &= \mathrm{softmax}\left(\frac{\boldsymbol{q}_t \boldsymbol{K}^T}{\sqrt{d_k}}\right) \end{aligned} \quad (1)$$

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

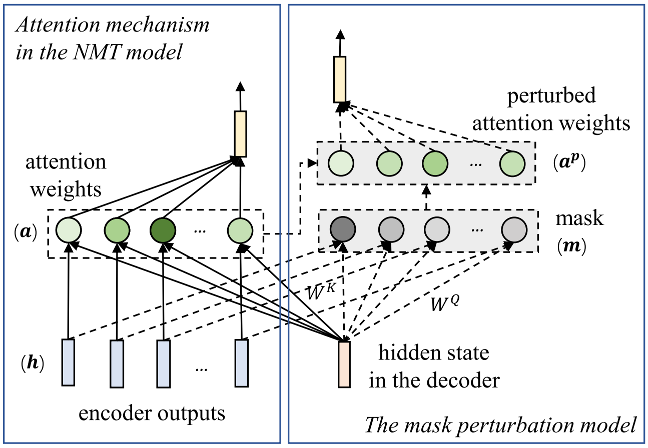

Attention mechanism in the NMT model
attention weights
(a)
...
(h)
...
encoder outputs
perturbed attention weights
...
$(\boldsymbol{a}^p)$
mask
(m)
...
W^K
W^Q
hidden state in the decoder
The mask perturbation model

Fig. 2. Overview of the mask perturbation model. It is trained to perturb the attention weights of decisive inputs to harm the performance, which is used as the analytical tool in Section III. The dashed arrow lines represent the operations in the mask perturbation model.

where d_k is the dimension of the keys. For MHA, we use different projections to obtain the queries, keys, and values representations for each head.

It is noted that the Transformer model performs N cross-lingual attention layers and employs h parallel attention heads for each time. Thus we implement our methods on $N \times h$ attention operations separately. For simplicity, we next denote the query, keys, and values as $\boldsymbol{q}_t, \boldsymbol{K}, \boldsymbol{V}$ regardless of what layers and heads they come from.

III. The Interpretability of Multi-Head Attention

This section explores whether multi-head attention in Transformer can explain inputs' contributions to the model output. We first propose a novel mask perturbation model to detect decisive inputs automatically and then compare them with high-attention tokens. We thoroughly analyze the extent to which the one with the highest attention weight leads the model to make its decision.

A. *Mask Perturbation Model*

To search the source-side inputs that the model relies on to produce the output, we can observe how the model prediction changes as perturbing different parts of the input sentence. We apply a mask to scale each input's attention weight, which simulates the process of perturbation.

As shown in Fig. 2, let $\boldsymbol{m}_t$ be a mask at t-th step. The **perturbed attention weight** $\boldsymbol{a}_t^p$ is calculated as:

$$\boldsymbol{a}_t^p = \boldsymbol{m}_t \odot \boldsymbol{a}_t + (\boldsymbol{1} - \boldsymbol{m}_t) \odot \boldsymbol{\mu}_0 \tag{2}$$

$\boldsymbol{\mu}_0$ is a uniform distribution (an average vector of $\frac{\boldsymbol{1}}{\boldsymbol{n}}$) and $\odot$ denotes element-wise multiplication. The mask $\boldsymbol{m}_t$ is obtained based on hidden states in the decoder $\boldsymbol{q}_t$ and keys $\boldsymbol{K}$:

$$\boldsymbol{m}_t = \sigma\left(\frac{\boldsymbol{q}_t \boldsymbol{W}^Q (\boldsymbol{K}\boldsymbol{W}^K)^T}{\sqrt{d_k}}\right) \tag{3}$$

Here, $\sigma(\cdot)$ is the sigmoid function. A smaller value of $\boldsymbol{m}_t$ means a larger perturbation extent on original attention weights. Considering the structure of multi-head attention in Transformer, $\boldsymbol{W}^Q$ and $\boldsymbol{W}^K$ differ among layers and heads.

To test the effect of perturbing different regions of inputs, we borrow the idea "deletion game" to find the smallest perturbation extent, which leads to a significant performance drop. The objective function of the mask perturbation model is:

$$\mathcal{L}(\theta^m) = -\mathcal{L}_{\text{NMT}}(\boldsymbol{a}_t^p, \theta) + \alpha \mathcal{L}_c(\theta^m) \tag{4}$$

where θ denotes the parameters of the original Transformer. $\mathcal{L}_{\text{NMT}}(\boldsymbol{a}_t^p, \theta)$ is the cross-entropy loss of the translation model when using perturbed attention weights $\boldsymbol{a}_t^p$. $\theta^m = \{\boldsymbol{W}^Q, \boldsymbol{W}^K\}$ represents parameters of the mask perturbation model. The first term indicates that the perturbation operation aims to degrade the translation quality. The second one serves as a penalty term to encourage most of the mask to be turned off (perturb inputs as few as possible).

$$\mathcal{L}_c(\theta^m) = \|\boldsymbol{1} - \boldsymbol{m}_t\|_2 \tag{5}$$

The perturbation extent is determined by the hyperparameter α. Under this setting, our mask perturbation model is trained to remove the most informative input to deteriorate the translation. The large perturbation extent indicates great contributions to the prediction and vice versa. We define the actual contribution of inputs at t-th step as the corresponding perturbation degree:

$$\boldsymbol{\varphi}_t = |\boldsymbol{a}_t^p - \boldsymbol{a}_t| \tag{6}$$

We aim to measure the extent to which actual contributions of inputs ($\boldsymbol{\varphi}_t$) are consistent with attention weights assigned to each token ($\boldsymbol{a}_t$). Considering that the greatest attention in $\boldsymbol{a}_t$ is often dominant (the value is often greater than 0.5), we simplify the question by comparing the top-1 in $\boldsymbol{\varphi}_t$ and $\boldsymbol{a}_t$.

$$\begin{aligned} i_t^* &= \arg\max_{i \in [1,n]} \varphi_{it} \\ j_t^* &= \arg\max_{j \in [1,n]} a_{it} \end{aligned} \tag{7}$$

At t-th step, the i_t^*-th input contributes most to the prediction, and the j_t^*-th input receives the highest attention weight. We further define *Reliability Level* as the likelihood that those two are the same.

$$Reliability\ Level = \frac{\sum_{t=1}^{|y|} \mathbb{1}\{i_t^* = j_t^*\}}{|y|} \tag{8}$$

We calculate this metric upon the whole dataset to evaluate the overall reliability level of attention weights. A high-reliability level means the token assigned with the highest attention weight is more likely to decide the model output.

Notably, earlier studies employ masks and "deletion game" as analytical tools to explore the importance of each attention head [15] or the contributions of each pixel in the figure to the model output [16]. Different from them, we extend to probing the inputs' contributions to the model prediction in NMT.

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

B. Experimental Setting

We present extensive analyses on three benchmarks: LDC Chinese ⇒ English (Zh ⇒ En),[1] WMT14 English ⇒ German (En ⇒ De)[2], and WMT14 English ⇒ French (En ⇒ Fr)[2]. We train the model on the training data and report analytical results on their validation set. For Zh ⇒ En, we remove sentences of more than 50 words and collect 2.1 M training samples. We use NIST 2002 as the validation set. For En ⇒ De, we train on 4.5 M training samples. For En ⇒ Fr, we extract 5.4 M sentence pairs from 35.8 M training corpora. We use newstest2014 dataset as the validation set for En ⇒ De and En ⇒ Fr.

To follow the standard NMT processing, we tokenize the corpora using a script from Moses [17]. We employ Byte Pair Encoding (BPE) [18] to all language pairs to construct a join vocabulary except for Zh ⇒ En where the source and target languages are separately encoded.

We experiment with `Base` Transformer [8] as default, which consists of a 6-layer encoder and 6-layer decoder and 8 attention heads. We find that the mask perturbation model tends to perturb the top-layer attention most, closer to the softmax layer. So our analyses are based on the top-layer attention. Following Garg *et al.* [19], we average eight heads to study the overall distribution of several attention heads. The NMT model and our mask perturbation model are jointly trained with 150 k steps, but their parameters are separately updated based on different loss functions. For the mask perturbation model, it works as an external detector to discover decisive inputs for each prediction, the gradients of which only flow to θ^m, as seen in Equation (4). Thus, it does not affect the optimization of the NMT model.

C. When Attention Weights Can Indicate Inputs' Contributions?

This section investigates the correlation between the reliability level of attention weights and the following properties: the entropy of attention weights, the syntactic role of the current translation, and the type of language pairs. Specifically, we first figure out the input contributing most to each prediction by our mask perturbation model. We then group target outputs based on the above three factors and calculate the overall reliability level for each group as in (8).

1) The Entropy of Attention Weights: To provide insights on the relation between the confidence level and the dispersion of attention distribution, we report the overall reliability level of attention weights where the entropy is no more than ε. The entropy of an attention distribution with n inputs are calculated as $e = -\sum_{i=1}^{n} a_i \text{log} a_i$, a metric to describe the dispersion of this distribution.

Observation 1: there is a negative correlation between the reliability level and the entropy of the attention distribution. As presented in Fig. 4, lines start at the peak at around 90% and gradually decline when the upper limit of entropy rises to 5.0. Taking Zh ⇒ En as an example, the attention can be relied on to explain inputs' contributions at the 90% confidence when its entropy is smaller than 1.5. However, the reliability level dramatically reduces to 47% as the entropy goes to 3.0.

Based on this, we can give detailed advice on setting the threshold according to the acceptable reliability level and language pairs. If we suppose the score is 0.7 in En ⇒ De task, the threshold should be no more than 3.0.

It is noted that a high entropy never means the distribution is uniform. It is hard for us to directly figure out the unreliable attention weights without any additional tool. From this, our model works as a detector to measure the reliability of attention. We further find a natural indicator, the distribution entropy, which can approximately evaluate the reliability of attention without external judgment.

2) The Syntactic Role of the Current Translation: In this experiment, we investigate the relation between the reliability level of attention weights and syntactic roles of the current translation. Words in German and French sentences are labeled by `nltk`,[3] and Chinese sentences are labeled by `jiaba`.[4] We distinguish between the following POS tags: noun, verb, adjective, adverb, number, preposition, conjunction, determiner, and punctuation. The first five tags belong to content words, and the others belong to function words. The results are displayed in Fig. 3, where the entropy of attention distribution is also given to provide a detailed comparison.

Observation 2: the ascending order of POS tags is similar among language pairs. The numeral is always the best, and noun, verb, and adjective are not far behind. Conjunction, punctuation, and preposition consistently rank at the bottom. Besides, we find nouns and verbs usually rank in the front of conjunction and preposition. It implies that the percentage of deceptive attention weights in function words is more significant than content words.

Thus, we suggest treating attention differently for different POS tags. As for numeral and adjective, the most attended word is particularly the one that decides the model prediction, and attention absolutely can be a reliable explanation. Nevertheless, the perturbation model sometimes does not heavily attack the highest weight, including conjunction and preposition. In this case, it is questionable to use attention weight as the indicator of inputs' contributions.

However, "⟨EOS ⟩" is the only exception, which is middle in Zh⇒ En and En ⇒ Fr but takes the last place in En ⇒ De. This is because our perturbation model finds it vain to pose any disturbance on source side to hamper the generation of "⟨EOS ⟩," which mostly relies on the target-side decoder. Only 6% of "⟨ EOS⟩" is attacked by mask perturbation model, the small scale of which leads to unstable results.

3) The Type of Language Pair: In addition to the two above factors, we also analyze from the view of language pairs. Among three translation tasks, Chinese belong to the Sino-Tibetan language family. English, French, and German are all Indo-European with overlapped vocabulary.

Observation 3: the overall confidence level varies across three language pairs. The general confidence level is relatively lower

[1] The corpora includes LDC2000T50, LDC2002T01, LDC2002E18, LDC2003E07, LDC2003E14, LDC2003T17 and LDC2004T07.

[2] [Online]. Available: http://www.statmt.org/wmt14/translation-task.html

[3] [Online]. Available: https://www.nltk.org/

[4] [Online]. Available: https://pypi.org/project/jieba/

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

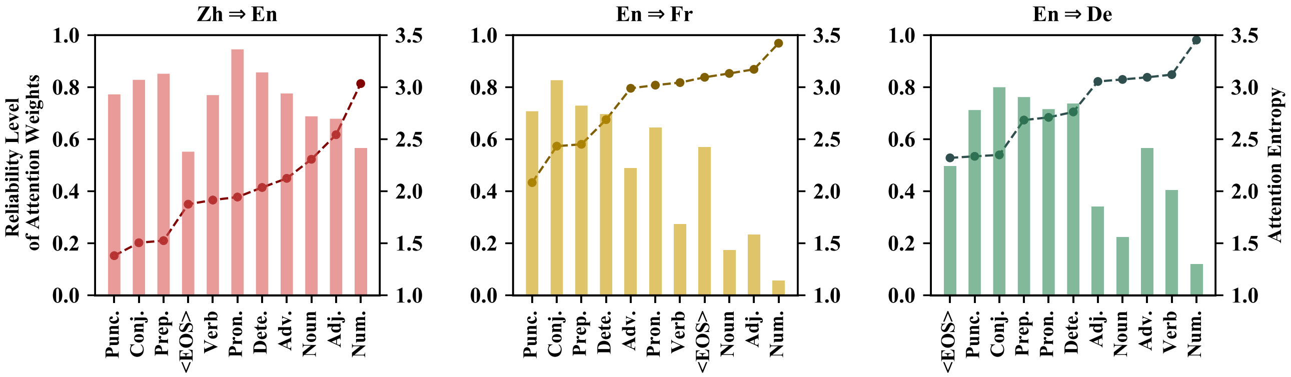


Fig. 3. Reliability level of attention weights when generating translation with different syntactic roles. The lines represent the reliability level. The bars illustrate entropies of attention weights in each group. The target-side POS tags are sorted by the reliability level of their corresponding attention weights.

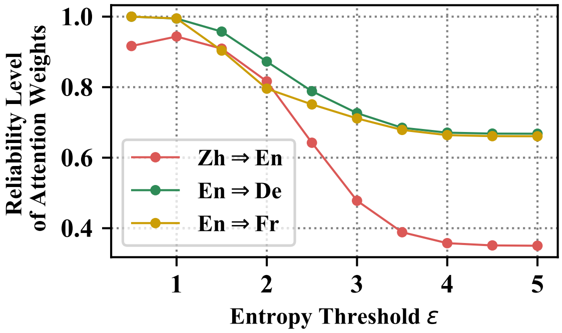


Fig. 4. Overall reliability level of attention weights where the entropy is no more than ε.

in Zh ⇒ En than those in En ⇒ Fr and En ⇒ De. It can attribute to the significant syntactical discrepancy between source and target language in the Zh ⇒ En task.

As shown in Fig. 3, there are around two-thirds of POS tags having the above-0.6 confidence level in En ⇒ Fr and En ⇒ De. Meanwhile, only numerals and adjectives meet that condition in Zh ⇒ En. Besides, the value of the attention entropy in Zh ⇒ En (indicated by bars in Fig. 3) is larger than those in the two other language pairs. This further adds evidence to the unreliability of attention when the Transformer performs Zh ⇒ En translation task.

A similar pattern is also found in Fig. 4. The figure experiences a rapid fluctuation in Zh ⇒ En translation rather than a gradual decrease in another two. When the threshold is larger than 3.0, attention is too weak to explain input tokens' contributions in the Zh ⇒ En task.

D. Revisiting the Divergence Between Attention Weights and Word Alignment

Many pieces of research reveal that attention weights extracted from the NMT model diverge from the word alignment between the source and target sentences. As shown in Section III-C, learned attention weights sometimes could not indicate actual source-target correspondence learned by the model. Thus, we tend to investigate whether unreliable attention weights worsen this divergence.

We first separately select the most significant input for each prediction based on attention weights and our measurement as described in (6). Due to BPE operation, two words are aligned if part of them are strongly connected. Then, we employ fast-align[5] [20] to obtain word alignment in parallel sentences as silver references. We use the *alignment error rate* (AER) [21] as the measurement of alignment quality. AER=0 represents a perfect consistency with word alignment.

Fig. 5 exhibits AER differences of attention weights ($\mathrm{AER}_{\mathrm{attn}}$) and our measurements ($\mathrm{AER}_{\mathrm{ours}}$). We see that alignments extracted from our measured inputs' contributions are much closer to word alignment, especially in En ⇒ Fr and En ⇒ De. The most striking AER differences are in conjunction and preposition, where attention weights are found to be most unreliable in Fig. 3. That is to say, the actual model manners are closer to word alignment but misled by unreliable attention weights.

Besides, we discover a different pattern in the Zh ⇒ En task, where the alignment obtained based on our measurements sometimes differs from word alignment to a greater degree when predicting nouns and verbs. In such cases, the model itself does not work as our expected word alignment to complete translation.

Note that we never mean the model behaviour in NMT should keep pace with the standard word alignment because attention is not alignment. The black box probably learns more unintelligible clues from the mass of data [22]. However, the comparisons aim to show how much we could understand model manners rather than establish a standard to score the quality of explanations. Thus, a stronger correlation with word alignment is not evidence for the superiority of our model. Our purpose is to show whether our measurements correspond to our intuition or not, and how to explain model behaviors from this new perspective.

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

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

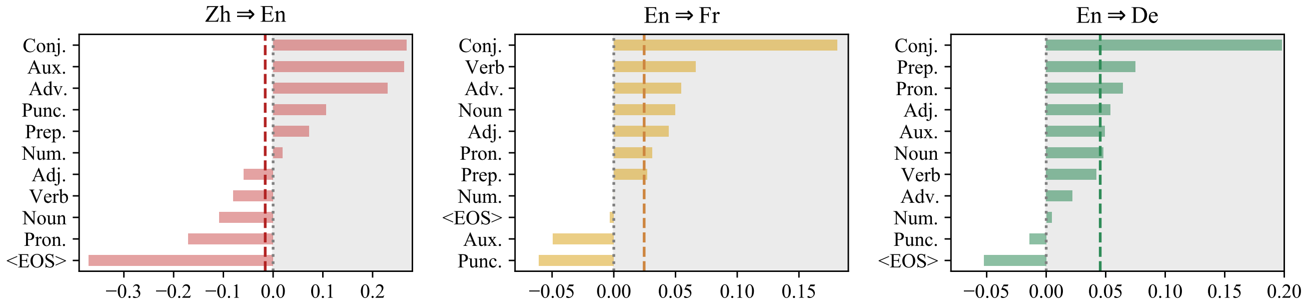


Fig. 5. AER differences of attention weights and our measured inputs' actual contributions for different syntactic roles ($\mathrm{AER}_{\mathrm{attn}} - \mathrm{AER}_{\mathrm{ours}}$). If the bar lies in grey areas ($\mathrm{AER}_{\mathrm{attn}} > \mathrm{AER}_{\mathrm{ours}}$), alignment extracted from our measurement is closer to word alignments between input and output words. The dashed lines represent average AER differences.

IV. Attention Calibration for Transformer

Based on observations in Section III, we can infer that attention weights in Transformer cannot always indicate inputs' contributions to the model output. In other words, attention mechanisms are incapable of precisely identifying decisive inputs for each prediction. Some unimportant words (e.g., punctuations) frequently attract high attention [23], resulting in wrong-translation or over-translation in NMT [13]. It further motivates us to calibrate the attention weights to focus more on decisive inputs to enhance translation accuracy.

A. Attention Calibration Network

The following operations are based on the mask perturbation model described in Section III-A. As aforementioned, our mask perturbation model removes the most informative input to deteriorate the translation by setting the corresponding mask to zero. In other words, a smaller mask means a larger perturbation, namely a more significant impact on the prediction. Thus, we propose to calibrate original attention weights in NMT by highlighting essential inputs for each model prediction, some of which are undiscovered under the attention mechanism.

Formally, the **calibrated attention weight** $\boldsymbol{a}_t^c$ can be designed as:

$$\boldsymbol{a}_t^c = \boldsymbol{a}_t \odot e^{1-\boldsymbol{m}_t} \tag{9}$$

We increase attention weights of key inputs which suffer from large perturbation extents. The attention weights of other unimportant inputs are correspondingly decreased. We design three methods to incorporate $\boldsymbol{a}_t^c$ into the original one $\boldsymbol{a}_t$ to obtain **combined attention weights** $\boldsymbol{a}_t^{\boldsymbol{comb}}$:

- **Fixed Weighed Sum**. In this method, the calibrated attention weights are added to the original attention weights of fixed ratio λ as:

$$\boldsymbol{a}_t^{\boldsymbol{comb}} = \mathrm{softmax}(\boldsymbol{a}_t + \lambda * \boldsymbol{a}_t^c) \tag{10}$$

- **Annealing Learning**. Considering the mask perturbation model is not well-trained at the early stage, we expect the effect of $\boldsymbol{a}_t^c$ to be smaller at first and gradually grow with the training step s. To this end, we use annealing learning to control the ratio of $\boldsymbol{a}_t^c$ as:

$$\begin{aligned} \boldsymbol{a}_t^{\boldsymbol{comb}} &= \gamma(s) * \boldsymbol{a}_t + (1 - \gamma(s)) * \boldsymbol{a}_t^c \\ \gamma(s) &= e^{-s/10^5} \end{aligned} \tag{11}$$

- **Gating Mechanism**. We propose a calibration gate to dynamically select the amount of the information from the perturbation model in the decoding process.

$$\begin{aligned} \boldsymbol{a}_t^{\boldsymbol{comb}} &= g_t * \boldsymbol{a}_t + (1 - g_t) * \boldsymbol{a}_t^c \\ g_t &= \sigma(\boldsymbol{q}_t \boldsymbol{W}^g + \boldsymbol{b}^g) \end{aligned} \tag{12}$$

  where $\boldsymbol{W}^g$ and $\boldsymbol{b}^g$ are trainable parameters varying among different layers and heads.

Our mask perturbation model and NMT model are jointly optimized. As shown in Fig. 6, the mask perturbation model is trained to worsen the performance by limited perturbation on attention weights (as seen in Equation (4)). Given what inputs are perturbed, we can figure out decisive inputs for each model prediction and calibrate original attention weights in the NMT model by ACN. With calibrated attention weights, the NMT model is finally optimized by:

$$\mathcal{L}_{\mathrm{NMT}}(\theta) = -\sum_{t=1}^{m} \log p(y_t | y_{<t}, x; \boldsymbol{a}_t^{\boldsymbol{comb}}, \theta) \tag{13}$$

During testing, the mask perturbation model also helps identify informative inputs based on the hidden state in the decoder at each step (as seen in Equation (3)). The NMT model decodes with calibrated attention weights. Moreover, our method can provide the saliency map between inputs and outputs based on the generated mask, an accessible measurement of inputs' contributions to model predictions.

B. Experimental Setting

Dataset. We assess our method in LDC Chinese-English (Zh ⇒ En), WMT14 English-German (En ⇒ De), WMT17 English-Latvian (En ⇔ Lv), WMT17 English-Finnish (En ⇔ Fi), and WMT16 English-Romanian (En ⇔ Ro).

We tokenize the corpora and apply BPE as mentioned in Section III-B. For Zh ⇒ En, we remove sentences of more than 50 words. We use NIST 2002 as validation set, NIST 2003-2006 as the test set. For En ⇒ De, newstest2013 and newstest2014

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

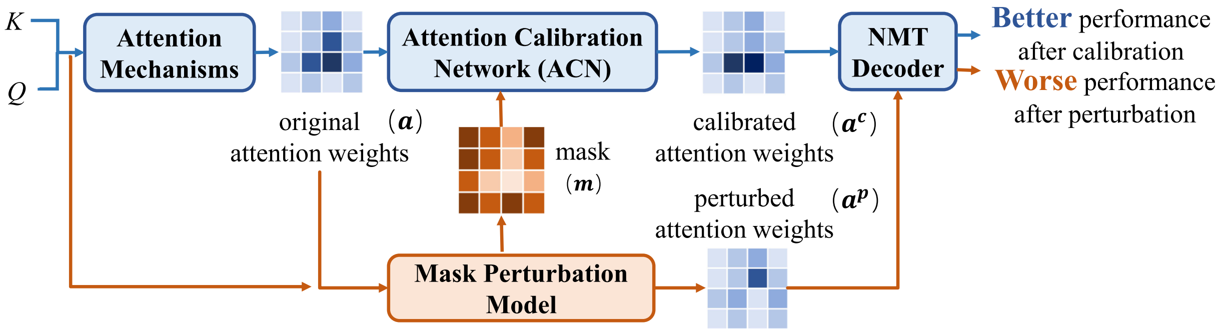

K
Q
Attention Mechanisms
original ($\boldsymbol{a}$) attention weights
Attention Calibration Network (ACN)
mask ($\boldsymbol{m}$)
calibrated ($\boldsymbol{a^c}$) attention weights
perturbed ($\boldsymbol{a^p}$) attention weights
Mask Perturbation Model
NMT Decoder
Better performance after calibration
Worse performance after perturbation

Fig. 6. Overview of the framework. The mask perturbation model is trained to perturb the attention weights of decisive inputs to harm the performance (as denoted by yellow arrows). ACN looks for what inputs are perturbed and enhance the corresponding attention weights (as shown by blue arrows), which is described in Section IV.

TABLE I
STATISTICS OF THE DATASETS

Source	Lang.	Train	Dev.	Test	Vocab.
LDC[1]	Zh⇒En	2.09M	878	4789	32k
WMT14[2]	En⇒De	4.54M	3000	3003	37k
WMT17[3]	En⇒Lv Lv⇒En	4.46M	2003	2001	37k
	En⇒Fi Fi⇒En	2.63M	3000	3002	32k
WMT16[4]	En⇒Ro Ro⇒En	0.61M	1999	1999	32k

[1] The corpora includes LDC2000T50, LDC2002T01, LDC2002E18, LDC2003E07, LDC2003E14, LDC2003T17 and LDC2004T07. Following previous work, we use case-insensitive tokenized BLEU to evaluate the performance.
[2] http://www.statmt.org/wmt14/translation-task.html
[3] http://www.statmt.org/wmt17/translation-task.html
[4] http://www.statmt.org/wmt16/translation-task.html

TABLE II
COMPARISON OF OUR MODEL, TRANSFORMER BASELINES AND RELATED WORK ON THE WMT14 EN ⇒ DE USING CASE-SENSITIVE BLEU. RESULTS WITH ‡ ARE TAKEN FROM CORRESPONDING PAPERS

Model		TEST
GNMT [25]‡		24.61
Conv [26]‡		25.16
AttIsAll [8]‡		27.30
FENMT [27]‡		27.70
Our Implemented Baseline		27.37
Ours	Fixed	27.38
	Anneal	**28.10**
	Gate	27.75

TABLE III
EVALUATION OF TRANSLATION QUALITY FOR ZH ⇒ EN TRANSLATION USING CASE-INSENSITIVE BLEU SCORE

Model		DEV	MT03	MT04	MT05	MT06	ALL
Baseline		48.56	49.58	48.58	49.95	47.22	48.24
Ours	Fixed	48.42	49.41	48.56	50.32	47.89	48.44
	Anneal	48.22	49.73	48.85	**50.97**	47.49	48.74
	Gate	**49.52**	**50.42**	**49.16**	50.78	**47.98**	**49.00**

are set as validation and test sets. We use the standard 4-gram BLEU [24] on the true-case output to score the performance. For En ⇔ Ro, we use newsdev2016 and newstest2016 as development and test sets. For En ⇔ Lv and En ⇔ Fi, newsdev2017 and newstest2017 are validation set and test set. See Table I for statistics of the data.

Settings. We implement described models with fairseq[6] toolkit for training and evaluating. We experiment with Transformer `Base` [8]: hidden size $d_{model} = 512$, 6 encoder and decoder layers, 8 attention heads and 2048 feed-forward inner-layer dimension. The dropout rate of the residual connection is 0.1 except for Zh ⇒ En (0.3). During training, we use label smoothing of value $\epsilon_{ls} = 0.1$ and employ the Adam ($\beta_1 = 0.9, \beta_2 = 0.998$) for parameter optimization with a scheduled learning rate of 4,000 warm-up steps. All experiments last for 150 k steps except for small-scale En ⇔ Ro translation tasks (100 k). For testing, we average the last ten checkpoints and use beam search (beam size 4, length penalty 0.6) for inference.

Besides, the hyperparameter λ in Equation (10) decides how much calibrated attention weights are incorporated in the Fixed Weighted Sum method. We set $\lambda = 0.1$ in all experiments.

C. Main Results

To comprehensively compare with existing baselines and similar work, we report the results of some competitive models including GNMT [25], Conv [26] and AttIsAll [8] on WMT14 En ⇒ De translation task. Besides, we also compare our method against the most related one, FE NMT, which introduces word alignment information to guide translation [27]. As presented in Table II, our method exhibits better performance than the above models. Unlike supervised attention with external word alignment, our model yields a significant gain by looking into what inputs affect the model's internal training.

Table III shows the translation quality measured in BLEU scores for NIST Zh ⇒ En. Our proposed model significantly

[6] [Online]. Available: https://github.com/pytorch/fairseq

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
Evaluation of Translation Quality for WMT17 En ⇔ Fi, WMT17 En ⇔ Lv and WMT16 En ⇔ Ro Using Case-Insensitive BLEU Score

Model		En⇒Lv	Lv⇒En	En⇒Fi	Fi⇒En	En⇒Ro	Ro⇒En
Baseline		16.26	17.76	22.01	26.07	22.56	27.53
Ours	Fixed	16.54	18.45	22.42	26.20	23.10	28.02
	Anneal	16.35	18.12	22.40	26.39	23.27	28.20
	Gate	**16.83**	**18.71**	**22.55**	**26.67**	**24.00**	**28.48**

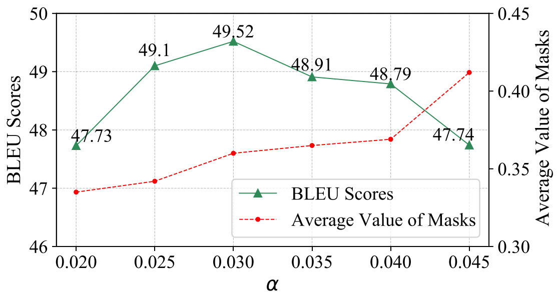


Fig. 7. Translation performance (BLEU score) on the validation set and average value of generated masks with respect to different hyperparameter α on Zh ⇒ En translation task (Gate Mechanism).

outperforms the baseline by 0.96 (MT02), 0.84 (MT03), 0.58 (MT04), 1.02 (MT05) and 0.76 (MT06), respectively.

We also conduct our experiments on WMT17 En ⇔ Fi and En ⇔ Lv. As shown in Table IV, our methods improve the performance over baseline by 0.54 BLEU (En ⇒ Fi), 0.6 BLEU (Fi ⇒ En), 0.57 BLEU (En ⇒ Lv) and 0.95 BLEU (Lv ⇒ En). For the small-scale WMT16 En ⇔ Ro, our methods achieve a substantial improvement of 1.44 more BLEU (En ⇒ Ro) and 0.95 BLEU (Ro ⇒ En). Compared to the large-scale dataset, the insufficient training data make it harder to learn the relationship between inputs and outputs, leaving a greater need for calibrating attention weights.

Overall, our proposed model significantly outperforms the strong baselines, especially for the small-scale dataset. More importantly, the parameter size is tiny (6 M), which does not add much cost to the training and inference process.

Effect of Fusion Methods. For three fusion methods, the fixed weighted sum has a limited gain. Annealing learning is comparatively more stable, which reduces the impact of ACN when the mask perturbation model is not well-trained in the initial. But it is challenging to design an annealing strategy that can be applied to all language pairs. Gate mechanism mostly achieves the best performance for dynamically controlling proportions of original and calibrated attention weights.

Effect of Hyperparameter. As shown in Equation (4), the hyperparameter α in the loss function of the mask perturbation model decides how much masks would turn on to perturb the original attention weights. Fig. 7 exhibits the average value of generated masks across heads as the function of the setting of α. A larger α forces the model to turn off most masks, which makes the value of the mask closer to 1, resulting in a smaller perturbation extent on the attention weights.

TABLE V
JSD Between Attention Weights Before and After Calibration At Each Layer on Zh ⇒ En and En ⇒ De Tasks. ↓ Denotes the JSD At One Layer is Lower Than Averaged JSD, While ↑ is Above Average. Note That the Overall JSD for Each Language Pair is Decided by the Hyperparameter α, But the Calibration Extents of Different Layers are Learned by ACN

Layer	Zh⇒En	En⇒De
1	0.0249 ↓	0.0784 ↑
2	0.0268 ↓	0.0695 ↑
3	0.0243 ↓	0.0708 ↑
4	0.0296 ↑	0.0581 ↓
5	0.0295 ↑	0.0428 ↓
6	0.0276 ↑	0.0548 ↓

D. Analysis

In this section, we explain how our proposed method helps produce better translation by investigating: (1) what attention weights need to calibrate and (2) calibrated attention weights are more focused or more uniform. Specifically, we delve into the differences between layers, which give insights into the attention mechanism's inner working. We conduct analyses on Zh ⇒ En NIST03 and En ⇒ De newstest2014 to understand our model from different perspectives.

We apply Jensen-Shannon Divergence (JSD) between attention weights before and after calibration to measure the calibration extent:

$$\mathrm{JSD}\left(a_1, a_2\right) = \frac{1}{2}\mathrm{KL}[a_1 \| \overline{a}] + \frac{1}{2}\mathrm{KL}[a_2 \| \overline{a}] \qquad (14)$$

where $\overline{a} = \frac{a_1 + a_2}{2}$. A high JSD means calibrated attention weights are distant from the original one. Besides, we use entropy changes of attention weights ($\triangle$Ent) to test whether calibrated attention weights become more uniform or focused.

$$\triangle\mathrm{Ent}\left(a_1, a_2\right) = e\left(a_1\right) - e\left(a_2\right) \qquad (15)$$

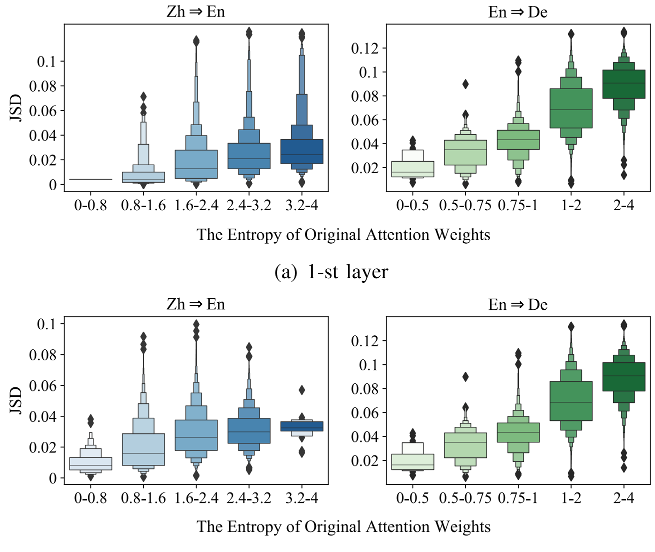
where $e(a) = -\sum_{i=1}^{m} a_i \log a_i$, a metric to describe the uncertainty of the distribution.

1) What attention weights need to calibrate?

High or low layers? Regarding distinct roles of multiple attention layers, one natural question is what attention layers are not well-trained in the original NMT model and urgently need to calibrate. Table V depicts the JSD between original and calibrated attention weights. We find JSD at 4-6 layers are greater than those at 1-3 layers in the Zh ⇒ En task, which means high-layer attention needs more calibration. However, there is a different pattern in the En ⇒ De task, where JSD at the high layer is smaller than at the low layers. We speculate that the difference is due to the language discrepancy, and we will explore this phenomenon in our future work.

High or low entropy? A lower entropy of attention weights suggests that several input tokens count for a large amount of attention. That is, the model is confident about its selection of important tokens [28]. We attempt to validate whether attention weights are more likely to be calibrated when the NMT model is uncertain about its decision. Fig. 8 displays a positive relationship between calibration extent and the entropy of attention weights. Take the 6-th attention layer in Zh ⇒ En translation as

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.



(a) 1-st layer

(b) 6-th layer

Fig. 8. JSD between attention weights before and after calibration with respect to the entropy of original attention distributions.

TABLE VI
Entropy Differences (△Ent) Between the Original and Calibrated Attention Weights. "+" Means the Calibrated Attention Weights are More Disperse. "-" Indicates Attention Weights are Sharper After Calibration

Layer	Zh⇒En	En⇒De
1	+ 0.0203	+ 0.1846
2	- 0.011	+ 0.0762
3	- 0.0023	+ 0.0207
4	- 0.0224	- 0.0336
5	- 0.0303	- 0.0595
6	- 0.0083	- 0.01
All	- 0.0336	- 0.0224

an example (as seen in Fig. 8(b)). The average JSD is 0.0084 for attention weights in rang [0,0.8], while the value is 0.0324 for attention weights where the entropy is larger than 3.2. These findings can also be observed at different attention layers and language pairs.

We infer that a higher entropy indicates the NMT model relies on multiple inputs to generate the translation, which increases the probability of information redundancy or error signals. Our proposed model is more prone to calibrate these attention weights, making the NMT model pay more attention to informative inputs.

2) Calibrated attention weights are more dispersed or focused?

We also explore underlying reasons why calibrated attention can boost performance from the perspective of entropy changes. We present the entropy differences of the original and calibrated attention weights in Table VI. We notice that entropies of attention weights are overall smaller after calibration. But, differences vary across layers. For En ⇒ De task, calibrated attention weights are more uniform at 1-3 layers and more focused at 4–6 layers, while attention weights become more concentrated at all layers except the 1-st layer in Zh ⇒ En task. These findings show that each attention layer plays a different role in the decoding process. Low layers generally grasp information from various inputs, while high layers look for particular words tied to model predictions.

V. Application to Text Summarization

We further verify the superiority of our approach on the abstractive text summarization task, which is another real-world application that attention mechanism succeeds [29]–[32]. This task aims to generate a summary with a few sentences that contain the primary information of an article, which benefits a lot from the attention mechanism to search salient ideas of the original.

A. Experimental Setting

Dataset. We use the *CNN/Daily Mail* dataset [29], which contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average). We download non-anonymized version of the data[7] using scripts supplied by [33], which has 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs. We use a sharing vocabulary of 50 k words for source and target side. We use the standard ROUGE as our evaluation metric [34], reporting the F1 scores for ROUGE-1, ROUGE-2 and ROUGE-L by the pyrouge package.[8]

Settings. We experiment with 4-layer Transformer with 8 attention heads. The dropout rate of the residual connection is 0.2. We train using Adagrad [35] with learning rate of 0.15 and an initial accumulator value of 0.1. All experiments last for 150 k steps. For testing, we average the last ten checkpoints and use beam search with beam size 4.

B. Results and Analysis

As shown in Table VII, we list the results of three competitive models, which are all equipped with the traditional attention mechanism. It is worth noting that pointer-generator [36] achieves better performance by using the attention mechanism in two ways: (1) dynamically fetching the relevant piece of information for generating the next word as usual, and (2) locating a certain segment of the input article and directly copying the segment to the output sequence. For comparison, we also add CopyNet [37] to Transformer model as our second baseline. In practice, we select one attention head from the 4-th decoder layer to find the copy words.

Compared with baseline I, calibrated attention weights help generate more precise summaries, and annealing learning performs the best. When implemented with CopyNet, attention calibration obtains significant gains over baseline II. We find that even the most simple fusion method does well. The fixed mode achieves +1.56 ROUGE-1, +0.79 ROUGE-2, and +1.39 ROUGE-L points, respectively. The considerable improvements

[7] [Online]. Available: https://github.com/abisee/cnn-dailymail

[8] [Online]. Available: https://pypi.org/project/pyrouge/0.1.3/

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
ROUGE Recall Evaluation Results on CNN/Daily Mail Dataset. Results With ‡ Mark are Taken From See *et al.* [33]. By Attention Calibration, Our Method Gets a Significant BLEU Improvement Than Baseline II

Model		RG-1	RG-2	RG-L
Seq-to-seq + attn [1]‡		31.33	11.81	28.83
Seq-to-seq + hierattn [31]		31.78	11.56	28.73
Pointer-generator [36]‡		36.44	15.66	33.42
Baseline I: Transformer		34.18	14.14	31.58
+ Calibrated attn	Fixed	34.46	14.26	31.81
	Anneal	34.77	14.45	32.10
	Gate	34.58	14.33	31.93
Baseline II: Transformer + CopyNet		38.24	17.26	35.23
+ Calibrated attn	Fixed	39.80	18.05	36.62
	Anneal	39.63	17.76	36.38
	Gate	39.17	17.48	36.03

are due to the enhanced copy accuracy yielded by calibrated attention weights.

The success of our calibration method in the text summarization task shows that attention weights' ability to detect salient inputs greatly affects the model performance in attention-based networks. It further reinforces the necessity of analyzing the interpretability of attention mechanisms and guiding attention weights to precisely locate essential inputs.

VI. Related Work

The attention mechanism is first introduced to augment vanilla recurrent network [1], [38], which are then the backbone of state-of-the-art Transformer [8] for NMT. It yields better performance and lets us have a closer look at how a model is operating [2], [4]. This section briefly introduce the taxonomy of attention mechanism and recent researches on analyzing and improving attention mechanisms.

A. The Taxonomy for Attention Mechanism

The attention mechanism allows for dynamically picking relevant parts in the input. It computes a weight distribution over input tokens and assigns higher values to more related ones. Galassi *et al.* [39] describe attention models based on the following orthogonal dimensions. (1) *The nature of inputs*. $\boldsymbol{K}$ and $\boldsymbol{V}$ are generally continuous representations of characters, words, or sentences. Li *et al.* [40] consider the inputs composed of texts and images. (2) *The compatibility function*. For example, dot function [38], scaled multiplicative function [8], convolution-based function [41] and so on. (3) *The distribution function*. It depends on the requirement of weights distribution, i.e., local or global attention [42], soft or hard attention [38]. (4) *Multiplicity*. It aims to generate multiple and heterogeneous inputs or outputs, which helps extract diverse information [43]. One typical example is the multi-head attention discussed in this article.

Considering the similar framework shared by different variants of the attention model, our work can easily apply to the above situations with adaption for specific structures.

B. Is Attention Interpretable?

Recent studies have spawned interest in whether attention weights faithfully represent each input token's responsibility for model prediction. Serrano and Smith [6] flip the model's decision by permuting some attention weights, which finds that high-weighted components not being the reason for the decision. Another line of work finds a weak correlation between attention scores and other well-ground feature importance metrics, specially gradient-based and leave-one-out methods, in various text classification tasks [44], [45]. Unlike criticizing attention weights as an explanation, Wiegreffe and Pinter [7] claim that trained attention mechanisms do learn something meaningful about the relationship between inputs and outputs, such as syntactic information [46], [47].

The above discussions are mainly done in the text classification with a focus on the classical attention function. We extend this question to Transformer-based NMT, which heavily relies on multi-head attention, an underexplored attention variant, to produce accurate translations. Unlike a "yes-or-no" answer given in prior work, we offer detailed advice on when attention is a reliable explanation for the model's behavior.

C. Does Attention Work as Word Alignment?

The key contribution of the attention model in NMT is the imposition of an alignment of the output to the input words. Arguably, Koehn *et al.* [11] state that the attention model for NMT does not always fulfill the role of a word alignment model, but may dramatically diverge, which is also found in Transformer-based NMT [12]. However, many works insist that NMT and word alignment are closely related tasks benefiting each other. Much effort has been made in guiding attention weights with word alignment [19], [48]–[51] or extract more accuracy word alignment from trained attention weights [12]. In this paper, we show that the discrepancy between attention model and word alignment in parallel sentences can be attributed to unreliable attention weights to a certain extent. In other words, the model beheviour is not much far from our intuition.

D. Can Attention Be Improved?

Many efforts have been made to strengthen the attention mechanism. They supervise attention weights with lexical probabilities [52], word alignment [19], [48]–[51], human rationales [53], and sparsity regularization [54]. Another related line of work locates important input to guide training in a self-supervised way [55]. Tang *et al.* [56] repeatedly extract active/misleading words based on attention weights which are used as attention supervision to retrain the model. Compared with them, our method is more efficient as done in a feed-forward manner. The contributions of input words are predicted by the mask perturbation model together with the training of the NMT model.

Another work line aims to make attention better indicative of the inputs' importance [23], [57]. They are designed for analysis with no significant performance gain, while we incorporate analytical results to enhance the NMT performance.

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

VII. Conclusion

In this paper, we have proposed a mask perturbation model to automatically discover decisive inputs for the model prediction and compare them with high-attention tokens. Thorough analytical results show that multi-head attention is not always a reliable indicator of inputs' contributions in Transformer-based NMT. We have introduced three factors related to the reliability level of attention: the entropy of attention weights, the syntactic role of the current translation, and the type of language pairs. We also discover that the discrepancy between attention weights and word alignment in the source and target sentence is partly due to unreliable attention weights.

Based on our findings, we propose three methods to calibrate the attention mechanism by focusing on discovered vital inputs. Extensive experiments on different-scale NMT tasks and text summarization tasks show that our approaches obtain significant improvements over the state-of-the-art system.

Attention mechanisms play different roles in varied tasks. In the future, we plan to measure the reliability level of attention distributions in more attention-based networks and further apply our attention calibration methods to more NLP applications (text classification, dialog system, and speech translation), and other attention-based computer vision tasks.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[2] Y. Belinkov and J. R. Glass, "Analysis methods in neural language processing: A survey," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 3348–3354.

[3] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable AI for natural language processing," in *Proc. 1st Conf. Asia-Pacific Chapter Assoc. Comput. Linguistics 10th Int. Joint Conf. Natural Lang. Process.*, 2020, pp. 447–459.

[4] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2020.

[5] S. Jain and B. C. Wallace, "Attention is not explanation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, vol. 1, pp. 3543–3556.

[6] S. Serrano and N. A. Smith, "Is attention interpretable," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, vol. 1, pp. 2931–2951.

[7] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *Proc. 2019 Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 11–20.

[8] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[9] Y. Chen, Y. Liu, G. Chen, X. Jiang, and Q. Liu, "Accurate word alignment induction from neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 566–576.

[10] Y. Cheng *et al.*, "Agreement-based joint training for bidirectional attention-based neural machine translation," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2761–2767.

[11] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proc. 1st Workshop Neural Mach. Transl.*, 2017, pp. 28–39.

[12] X. Li, G. Li, L. Liu, M. Meng, and S. Shi, "On the word alignment from neural machine translation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1293–1303.

[13] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, vol. 1, pp. 76–85.

[14] Y. Lu, J. Zeng, J. Zhang, S. Wu, and M. Li, "Attention calibration for transformer in neural machine translation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 1288–1298.

[15] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3449–3457.

[16] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, vol. 1, 2019, pp. 5797–5808.

[17] P. Koehn *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics* 2007, pp. 177–180.

[18] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.

[19] S. Garg, S. Peitz, U. Nallasamy, and M. Paulik, "Jointly learning to align and translate with transformer models," in *Proc. 2019 Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4452–4461.

[20] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proc. Hum. Lang. Technol.: Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2013, pp. 644–648.

[21] R. Mihalcea and T. Pedersen, "An evaluation exercise for word alignment," in *Proc. HLT-NAACL Workshop Building Using Parallel Texts: Data Driven Mach. Transl. Beyond*, 2003, pp. 1–10.

[22] H. Ghader and C. Monz, "What does attention in neural machine translation pay attention to," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, 2017, pp. 30–39.

[23] A. K. Mohankumar, P. Nema, S. Narasimhan, M. M. Khapra, B.V. Srinivasan, and B. Ravindran, "Towards transparent and explainable attention models," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4206–4216.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.

[25] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1243–1252.

[27] R. Weng, H. Yu, X. Wei, and W. Luo, "Towards enhancing faithfulness for neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2675–2684.

[28] E. Voita, R. Sennrich, and I. Titov, "Analyzing the source and target contributions to predictions in neural machine translation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguist. 11th Int. Joint Conf. Natural Lang. Process. (Volume 1: Long Papers)*, 2021, pp. 1126–1140.

[29] R. Nallapati, B. Zhou, C. N. dos santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.

[30] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. 2015 Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 379–389.

[31] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. 2016 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 93–98.

[32] J. Zhu, Y. Zhou, J. Zhang, and C. Zong, "Attend, translate and summarize: An efficient method for neural cross-lingual summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1309–1321.

[33] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1073–1083.

[34] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, pp. 74–81.

[35] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[36] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. NIPS'15 Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2692–2700.

[37] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1631–1640.

[38] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, L. Márquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds., 2015, pp. 1412–1421.

[39] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, pp. 4291–4308, Oct. 2021.

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.

[40] H. Li, J. Zhu, T. Liu, J. Zhang, and C. Zong, “Multi-modal sentence summarization with modality attention and image filtering,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 4152–4158.

[41] J. Du, L. Gui, Y. He, R. Xu, and X. Wang, “Convolution-based neural attention with applications to sentiment classification,” *IEEE Access*, vol. 7, pp. 27983–27992, 2019.

[42] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[43] J. Du, J. Han, A. Way, and D. Wan, “Multi-level structured self-attentions for distantly supervised relation extraction,” in *Proc. EMNLP*, 2018, pp. 2216–2225.

[44] S. Jain and B. C. Wallace, “Attention is not explanation,” in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, vol. 1, 2019, pp. 3543–3556.

[45] S. Vashishth, S. Upadhyay, G. S. Tomar, and M. Faruqui, “Attention interpretability across NLP tasks,” *CoRR*, 2019, *arXiv:1909.11218, 2019*.

[46] A. Raganato and J. Tiedemann, “An analysis of encoder representations in transformer-based machine translation,” in *Proc. EMNLP Workshop BlackboxNLP: Analyzing Interpreting Neural Netw. NLP*, 2018, pp. 287–297.

[47] J. Vig and Y. Belinkov, “Analyzing the structure of attention in a transformer language model,” in *Proc. ACL Workshop BlackboxNLP: Analyzing Interpreting Neural Netw. NLP*, 2019, pp. 63–76.

[48] W. Chen, E. Matusov, S. Khadivi, and J. Peter, “Guided alignment training for topic-aware neural machine translation,” in *Proc. 12th Conf. Assoc. Mach. Trans. Amer.*, S. Green and L. Schwartz, Eds., 2016, pp. 121–134.

[49] L. Liu, M. Utiyama, A. Finch, and E. Sumita, “Neural machine translation with supervised attention,” in *Proc. COLING, 26th Int. Conf. Comput. Linguistics: Tech. Papers*, 2016, pp. 3093–3102.

[50] H. Mi, Z. Wang, and A. Ittycheriah, “Supervised attentions for neural machine translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2283–2288.

[51] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, “Incorporating structural alignment biases into an attentional neural translation model,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 876–885.

[52] P. Arthur, G. Neubig, and S. Nakamura, “Incorporating discrete translation lexicons into neural machine translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1557–1567.

[53] J. Strout, Y. Zhang, and R. Mooney, “Do human rationales improve machine explanations,” in *Proc. ACL Workshop BlackboxNLP: Analyzing Interpreting Neural Netw. NLP*, 2019, pp. 56–62.

[54] J. Zhang, Y. Zhao, H. Li, and C. Zong, “Attention with sparsity regularization for neural machine translation and summarization,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 3, pp. 507–518, Mar. 2019.

[55] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6488–6496.

[56] J. Tang *et al.*, “Progressive self-supervised attention learning for aspect-level sentiment analysis,” in *Proc. ACL*, 2019, pp. 557–566.

[57] M. Tutek and J. Snajder, “Staying true to your word: (how) can attention become explanation,” in *Proc. 5th Workshop Representation Learn. NLP*, 2020, pp. 131–142.

Authorized licensed use limited to: Lappeenranta-Lahti University of Technology LUT. Downloaded on April 08,2024 at 12:45:22 UTC from IEEE Xplore. Restrictions apply.