Attention Calibration for Transformer in Neural Machine Translation

Yu Lu[1,2]*, Jiali Zeng[3], Jiajun Zhang[1,2]†, Shuangzhi Wu[3] and Mu Li[3]
[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Tencent Cloud Xiaowei, Beijing, China
{yu.lu, jjzhang}@nlpr.ia.ac.cn
{lemonzeng, frostwu, ethanlli}@tencent.com

Abstract

Attention mechanisms have achieved substantial improvements in neural machine translation by dynamically selecting relevant inputs for different predictions. However, recent studies have questioned the attention mechanisms' capability for discovering decisive inputs. In this paper, we propose to calibrate the attention weights by introducing a mask perturbation model that automatically evaluates each input's contribution to the model outputs. We increase the attention weights assigned to the indispensable tokens, whose removal leads to a dramatic performance decrease. The extensive experiments on the Transformer-based translation have demonstrated the effectiveness of our model. We further find that the calibrated attention weights are more uniform at lower layers to collect multiple information while more concentrated on the specific inputs at higher layers. Detailed analyses also show a great need for calibration in the attention weights with high entropy where the model is unconfident about its decision[1].

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

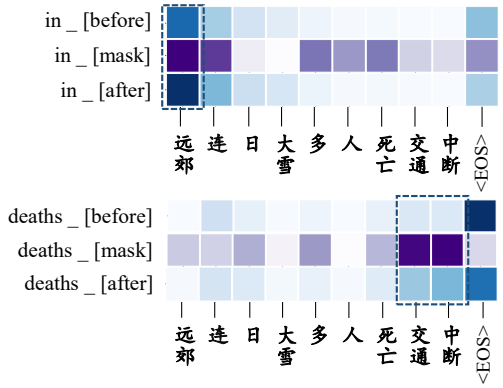


Figure 1: Examples of the attention weights before and after calibration. "in _" denotes the timestep after the prediction "in". The dashed boxes indicate the inputs which should receive more attention measured by our mask perturbation model.

1 Introduction

Attention mechanisms have been ubiquitous in neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017). It dynamically encodes source-side information by inducing a conditional distribution over inputs, where the ones that are most relevant to the current translation are expected to receive more attention.

However, many studies doubt whether highly-attended inputs have a large impact on the model outputs. On the one hand, erasing the representations accorded high attention weights do not necessarily lead to a performance decrease (Serrano and Smith, 2019), which can be attributed to that unimportant words (e.g., punctuations) are frequently assigned with high attention weights (Mohankumar et al., 2020). On the other hand, Jain and Wallace (2019) state that attention weights are inconsistent with other feature importance metrics in text classification tasks. It further proves that attention mechanisms are incapable of precisely identifying decisive inputs for each prediction, which would result in wrong-translation or over-translation in NMT (Tu et al., 2016). We take Figure 1 as an example. After producing the target-side word "deaths", attention mechanisms wrongly attribute most attention to the "$\langle EOS \rangle$", making parts of the source sentence untranslated.

In this paper, we propose to calibrate the vanilla attention mechanism by focusing more on key in-

*Work done while the author was an intern at Tencent.
†Corresponding author.
[1]https://github.com/yulu-dada/Attention-calibration-NMT

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 1288–1298
August 1–6, 2021.

puts. To test what inputs affect the model prediction most, we tend 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", which aims to find the smallest perturbation extents that cause the significant quality degradation. In this manner, we can find the most informative inputs for the prediction.

Based on the results detected by the mask perturbation model, we further calibrate attention weights by reallocating more attention to informative inputs. We design three fusion methods to incorporate the calibrated attention weights into original attention weights: (1) fixed weighted sum, (2) annealing learning, and (3) gating mechanism. The mask perturbation model and NMT model are jointly trained, while the attention weights in NMT are corrected based on the actual contributions measured by the mask perturbation model.

Recall the example in Figure 1. After producing the target word "in", our mask perturbation model finds that the source word "远郊 [countryside]" with a high attention weight is exactly the decisive input for the prediction. Therefore, we strengthen the corresponding attention weight of "远郊 [countryside]". However, after the prediction "deaths", the highly-attended "$\langle EOS \rangle$" is not the decisive input at the current step. We redistribute the attention weights to the source words ("交通 [traffic]" and "中断 [interruption]") which receive little attention but are important for the subsequent translation discovered by our mask perturbation model. After calibration, the missing source information "traffic interruption" is well-translated.

We conduct extensive experiments to verify our method's effectiveness on Transformer-based translation (NIST Zh⇒En, WMT14 En⇒De, WMT16 En⇔Ro, WMT17 En⇔Fi, and En⇔Lv). Experimental results show that our calibration methods can significantly boost performance. We further visualize calibrated attention weights and investigate when attention weights need to be corrected.

The contributions of this paper are three-fold:

- We propose a mask perturbation model to automatically assess each input's contribution for translation, which is simple yet effective.
- We design three methods to calibrate original attention weights by highlighting the informative inputs, which are experimentally proved to outperform strong baselines.
- Detailed analyses show that calibrated attention weights are more uniform at lower layers while more focused at the higher layers. High-entropy attention weights are found to have great needs for calibration at all layers.

2 Background

In this section, we first briefly introduce the framework of Transformer (Vaswani et al., 2017) with a focus on the Multi-head attention (MHA). Then we present an analysis of the learned attention weights, the correlation with feature importance measures, which motivates our ideas discussed afterward.

2.1 Transformer Architecture

The Transformer is an encoder-decoder framework with stacking layers of attention blocks. The encoder first transforms an input $x = \{x_1, x_2, ...x_n\}$ to a sequence of continues representations $\boldsymbol{h} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, ...\boldsymbol{h_n}\}$, from which the decoder generates an output sequence $y = \{y_1, y_2, ...y_m\}$.

Multi-head attention between encoder and decoder enables each prediction to attend overall inputs from different representation subspaces jointly. For the single head, we first project $\boldsymbol{h} = \{\boldsymbol{h_1}, \boldsymbol{h_2}, ...\boldsymbol{h_n}\}$ 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 an attention $\boldsymbol{a}_t$, which is used to calculate a weighted sum of values $\boldsymbol{V}$.

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

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 Transformer (base model) performs $N = 6$ cross-lingual attention layers and employs $h = 8$ 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.

2.2 Disagreement Between Attention Weights and Feature Importance Metrics

Attention mechanisms provide a distribution over the context representations of inputs, which are

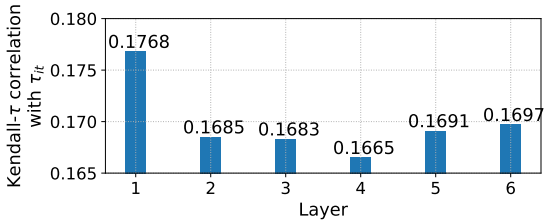


Figure 2: The mean **Kendall-τ** correlation between attention weights ($\boldsymbol{a}$) and gradient importance metrics (τ_{it}) on Zh⇒En translation.

often presented as communicating the relative importance of inputs. However, recent work has cautioned against whether the inputs accorded high attention weights decide the model outputs (Jain and Wallace, 2019). Our analysis examines the correlation with attention weights and feature importance metrics in NMT to test if the attention mechanisms focus on the decisive inputs. We apply gradient-based methods (Simonyan et al., 2014; Li et al., 2016) to measure the importance of each contextual representation h_i for model output y_t:

$$\tau_{it} = |\nabla_{h_i} p(y_t|\mathrm{x}_{1:\mathrm{n}})| \tag{2}$$

We train a baseline Transformer model on NIST Zh⇒En dataset and extract the averaged attention weights over heads.

Figure 2 reports the statistics of Kendall-τ correlation for each attention layer, where the observed correlation is all modest (0 indicates no correlation, while 1 implies perfect concordance). The inconsistency with feature importance metrics reveals that the high-attention inputs are not always responsible for the model prediction. It further motivates us whether we can calibrate the attention weights to focus more on the decisive inputs to achieve better translation.

3 Our Method

We aim to make the attention mechanism more focused on the informative inputs. The first step is to discover what inputs are essential for the model prediction. As shown in Figure 3, we design a **Mask Perturbation Model** to worsen the performance with limited perturbation on the original attention weights. By doing this, we can automatically detect what inputs decide the model outputs. Then, we design an **Attention Calibration Network (ACN)** to correct the original attention weights, highlighting the decisive inputs based on what inputs are perturbed by the mask perturbation model.

3.1 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.

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

$$\boldsymbol{a}_t^p = \boldsymbol{m}_t \odot \boldsymbol{a}_t + (\mathbf{1} - \boldsymbol{m}_t) \odot \boldsymbol{\mu}_0 \tag{3}$$

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

$$\boldsymbol{m}_t = \sigma\left(\frac{\boldsymbol{q}_t \boldsymbol{W}^Q (\boldsymbol{K}\boldsymbol{W}^K)^T}{\sqrt{d_k}}\right) \tag{4}$$

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 distinct regions of inputs, we borrow the idea "deletion game" to find the smallest perturbation extent, which leads to a significant performance decrease. The objective function of mask perturbation model is:

$$\mathcal{L}(\theta^m) = -\mathcal{L}_{\mathrm{NMT}}(\boldsymbol{a}_t^p, \theta) + \alpha \mathcal{L}_c(\theta^m) \tag{5}$$

where θ denotes the parameters of the original Transformer. $\mathcal{L}_{\mathrm{NMT}}(\boldsymbol{a}_t^p, \theta)$ 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 the parameters of mask perturbation model. The first term indicates that the perturbation operation aims to harm 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).

$$\mathcal{L}_c(\theta^m) = \|\mathbf{1} - \boldsymbol{m}_t\|_2 \tag{6}$$

The perturbation extent is determined by the hyperparameter α. Notably, earlier studies employ masks and "deletion game" as the analytical tools to explore the importance of each attention head (Fong and Vedaldi, 2017) or the contributions of the pixels in the figure to the model outputs (Voita et al., 2019). However, we extend to probing the inputs' contributions to the model prediction in NMT and further use the masks to calibrate the attention mechanisms based on the analytical results.

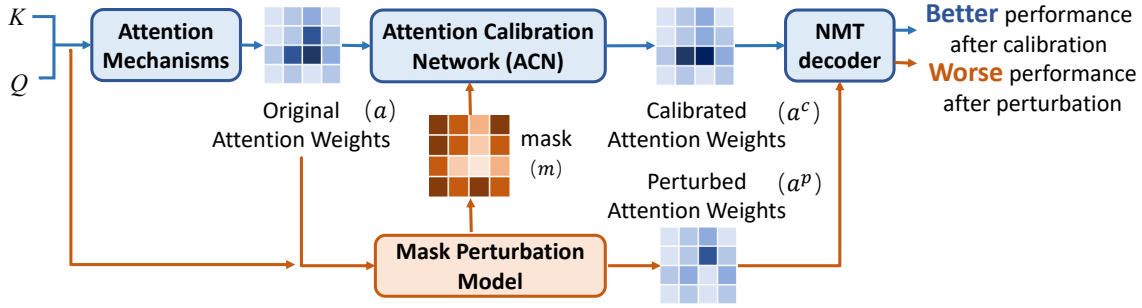


Figure 3: The overview of the framework. The mask perturbation model is trained to perturb the attention weights of decisive inputs to harm the performance. ACN looks for what inputs are perturbed and enhance the corresponding attention weights.

3.2 Attention Calibration Network

As aforementioned, our mask perturbation model removes the most informative input to deteriorate the translation by setting the corresponding masks to zero. In other words, a smaller mask means a larger perturbation, namely a more significant impact on the prediction. We propose to calibrate the original attention weights in NMT by highlighting the essential inputs for each model prediction.

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

$$\boldsymbol{a}_t^c = \boldsymbol{a}_t \odot e^{1-\boldsymbol{m}_t} \tag{7}$$

We increase the attention weights of key inputs which suffer large perturbation extents. The attention weights of other less-informative 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:

$$\boldsymbol{a}_t^{\boldsymbol{comb}} = \mathrm{softmax}(\boldsymbol{a}_t + \lambda * \boldsymbol{a}_t^c) \tag{8}$$

- **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:

$$\begin{aligned} \boldsymbol{a}_t^{\boldsymbol{comb}} &= \gamma(s) * \boldsymbol{a}_t + (1-\gamma(s)) * \boldsymbol{a}_t^c \\ \gamma(s) &= e^{-s/10^5} \end{aligned} \tag{9}$$

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

$$\begin{aligned} \boldsymbol{a}_t^{\boldsymbol{comb}} &= g_t * \boldsymbol{a}_t + (1-g_t) * \boldsymbol{a}_t^c \\ g_t &= \sigma(\boldsymbol{q}_t \boldsymbol{W}^g + \boldsymbol{b}^g) \end{aligned} \tag{10}$$

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

3.3 Training

Our mask perturbation model and NMT model are jointly optimized. As shown in Figure 3, the mask perturbation model is trained to worsen the performance by limited perturbation on the attention weights (Equation 5). Given what inputs are perturbed, we can figure out the decisive inputs for each model prediction and calibrate the original attention weights in the NMT model by ACN. With the calibrated attention weights, the NMT model is finally optimized by:

$$\mathcal{L}_{\mathrm{NMT}}(\theta) = -\sum_{t=1}^{m} \log p(y_t | y_{<t}, x; \boldsymbol{a}_t^{\boldsymbol{comb}}, \theta) \tag{11}$$

During testing, the mask perturbation model also helps identify the informative inputs based on the hidden state in the decoder at each step (as seen in Equation 4). The NMT model decodes with the calibrated attention weights. Moreover, our method can provide the saliency map between inputs and outputs based on the generated mask, an accessible measurement of the inputs' contributions to the model predictions.

4 Experiments

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

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

Table 1: Statistics of the datasets.

4.1 Dataset

We tokenize the corpora using a script from Moses (Koehn et al., 2007). Byte pair encoding (BPE) (Sennrich et al., 2016) is applied to all language pairs to construct a join vocabulary except for Zh⇒En where the source and target languages are separately encoded.

For Zh⇒En, we remove the sentences of more than 50 words. We use NIST 2002 as validation set, NIST 2003-2006 as the testbed. For En⇒De, newstest2013 and newstest2014 are set as validation and test sets. We use the standard 4-gram BLEU (Papineni et al., 2002) 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 1 for statistics of the data.

4.2 Settings

We implement the described models with fairseq[5] toolkit for training and evaluating. We experiment with Transformer Base (Vaswani et al., 2017): 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 the experiments last for 150k steps except for small-scale En⇔Ro translation tasks (100k). For evaluation, 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 8 decides how much the calibrated attention weights are incorporated in the Fixed Weighted Sum method. We set $\lambda = 0.1$ in all experiments for comparison.

[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

[5]https://github.com/pytorch/fairseq

Model		TEST
GNMT (Wu et al., 2016)‡		24.61
Conv (Gehring et al., 2017)‡		25.16
AttIsAll (Vaswani et al., 2017)‡		27.3
(Feng et al., 2020)‡		27.55
(Weng et al., 2020)‡		27.7
Our Implemented Baseline		27.37
Ours	Fixed	27.38
	Anneal	**28.1***
	Gate	27.75

Table 2: The comparison of our model, Transformer baselines and related work on the WMT14 En⇒De using case-sensitive BLEU. Results with ‡ mark are taken from the corresponding papers. "$*$" indicates the gains are statistically significant than baselines with $p<0.05$.

4.3 Main Results

To comprehensively compare with the existing baselines and similar work, we report the results of some competitive models including GNMT (Wu et al., 2016), Conv (Gehring et al., 2017) and AttIsAll (Vaswani et al., 2017) on WMT14 En⇒De translation task. Besides, we also compare our method against related researches about introducing word alignment information to guide translation (Weng et al., 2020; Feng et al., 2020). As presented in Table 2, 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 3 shows the translation quality measured in BLEU score for NIST Zh⇒En. Our proposed model significantly 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 4, 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). Com-

Model		DEV	MT03	MT04	MT05	MT06	AVE
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***

Table 3: Evaluation of translation quality for Zh⇒En Translation using case-insensitive BLEU score. "$*$" indicates the gains are statistically significant than baselines with $p<0.05$.

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	26.2	23.1	28.02
	Anneal	16.35	18.12	22.4	26.39	23.27*	28.2*
	Gate	**16.83***	**18.71***	**22.55***	**26.67***	**24.00***	**28.48***

Table 4: Evaluation of translation quality for WMT17 En⇔Fi, WMT17 En⇔Lv and WMT16 En⇔Ro using case-insensitive BLEU score. "$*$" indicates the gains are statistically significant than baselines with $p<0.05$.

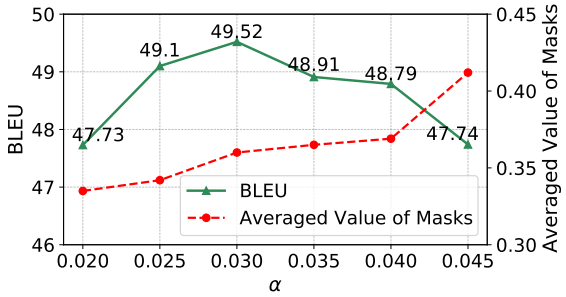


Figure 4: Experimental results on the validation set and the averaged value of generated masks with respect to different hyperparameter α on Zh⇒En translation task (Gate Mechanism).

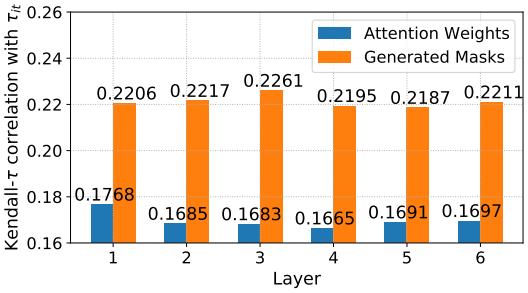


Figure 5: The mean **Kendall-τ** correlation between attention weights ($\boldsymbol{a}$), the masks ($\boldsymbol{m}$) generated by our mask perturbation model and gradient importance measures (τ_{it}) on Zh⇒En.

pared 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 (6M), which cannot 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 at the initial stage. 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 the proportions of original and calibrated attention weights.

Effect of Hyperparameter The hyperparameter α in the loss function of the mask perturbation model (as in Equation 5) decides how much masks would turn on to perturb the original attention weights. Figure 4 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.

Correlation with Feature Importance Metrics Figure 5 reports the correlation between our generated mask ($\boldsymbol{m}$) and the gradient-based importance measures[6] (τ_{it}). We find that the masks are relatively closer to the gradient-based importance measures than the original attention weights, which

[6]Though these measures are insufficient for telling what inputs are important (Kindermans et al., 2019), they do provide measures of individual feature importance with known semantics (Ross et al., 2017).

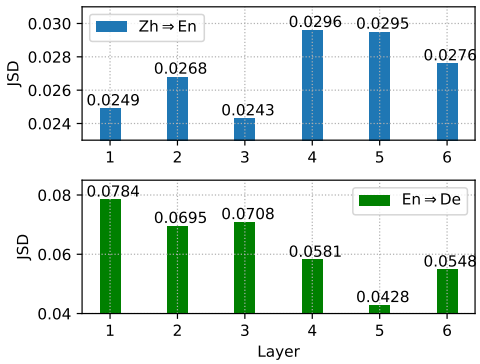
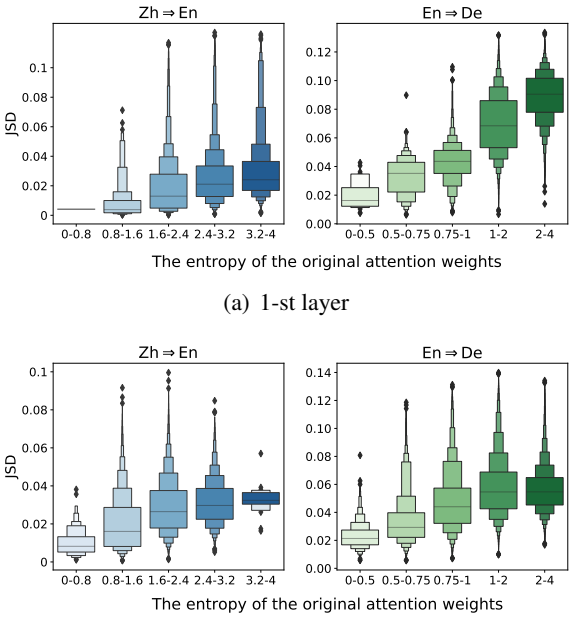


Figure 6: The JSD between attention weights before and after calibration at different layers on Zh⇒En and En⇒De translation. Note that the overall JSD for each language pair is decided by the hyperparameter α, but the calibration extents of layers are learned by ACN.



(a) 1-st layer

(b) 6-th layer

Figure 7: The JSD between attention weights before and after calibration with respect to the entropy of original attention distributions.

prove the effectiveness of our mask perturbation model to discover decisive inputs.

5 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:

$$\mathrm{JSD}\left(a_1, a_2\right) = \frac{1}{2}\mathrm{KL}[a_1 \| \overline{a}] + \frac{1}{2}\mathrm{KL}[a_2 \| \overline{a}] \quad (12)$$

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

$$\triangle\mathrm{Ent}\left(a_1, a_2\right) = \mathrm{ent}\left(a_1\right) - \mathrm{ent}\left(a_2\right) \quad (13)$$

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

5.1 What attention weights need to calibrate?

High or low layers? Concerning the roles of different attention layers, one natural question is what attention layers are not well-trained in the original NMT model and have an urgent need to calibrate. Figure 6 depicts the JSD between original and calibrated attention weights. We find high JSD for high layers and low JSD for low layers in Zh⇒En task. However, a different pattern is observed in En⇒De task, where JSD in the high layer is lower than in 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? More focused contributions of inputs suggest that the model is more confident about the choice of important tokens (Voita et al., 2020). We attempt to validate whether the attention weights are more likely to be calibrated when the NMT model is uncertain about its decision. Figure 7 shows the positive relationship between calibration extent and the entropy of attention weights. Take the 6-th attention layer in Zh⇒En translation as an example (as seen in Figure 7(b)). The averaged JSD is 0.0084 for the attention weights in rang [0,0.8], while the value is 0.0324 for the 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

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

Table 5: Entropy differences ($\triangle$Ent) between the original and calibrated attention weights. "+" means the calibrated attention weights are more disperse. "-" indicates attention weights are sharper after calibration.

translation, which increases the probability of information redundancy or error signals. Our proposed model is more likely to calibrate these attention weights to makes the NMT model pay more attention to the informative inputs.

5.2 Calibrated attention weights are more dispersed or focused?

There are multiple reasons why the calibrated attention weights can boost performance. Section 4.3 states that our generated masks are much closer to the gradient-based feature importance measures compared with attention weights. On the other hand, we present the entropy differences of the original and calibrated attention weights in Table 5 where the entropy of attention weights are overall smaller after calibration. However, the changes vary across layers. For En⇒De translation, the calibrated attention weights are more uniform at 1-3 layers and more focused at 4-6 layers, while the attention weights become more focused for all layers except the 1-st layer on Zh⇒En task. These findings prove that each attention layer plays a different role in the decoding process. The low layers generally grasp information from various inputs, while the high layers look for some particular words tied to the model predictions.

6 Related Work

The attention mechanism is first introduced to augment vanilla recurrent network (Bahdanau et al., 2015; Luong et al., 2015), which are then the backbone of state-of-the-art Transformer (Vaswani et al., 2017) for NMT. It yields better performance and provides a window into how a model is operating (Belinkov and Glass, 2019; Du et al., 2020). This section reviews the recent researches on analyzing and improving attention mechanisms.

The Attention Debate Many recent studies have spawned interest in whether attention weights faithfully represent each input token's responsibility for model prediction. Serrano and Smith flip the model's decision by permuting some small attention weights, with high-weighted components not being the reason for the decision. Some work (Jain and Wallace, 2019; Vashishth et al., 2019) find 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. We also present the correlation analysis in the less-discussed Transformer-based NMT and reach a similar conclusion. As opposed to the critiques of regarding attention weights as explanation, Wiegreffe and Pinter claim that the trained attention mechanisms do learn something meaningful about the relationship between inputs and outputs, such as syntactic information (Raganato and Tiedemann, 2018; Vig and Belinkov, 2019; Pham et al., 2019).

Can Attention be improved? There is plenty of work on supervising attention weights with lexical probabilities (Arthur et al., 2016), word alignment (Chen et al., 2016; Liu et al., 2016; Mi et al., 2016; Cohn et al., 2016; Garg et al., 2019; Feng et al., 2020), human rationales (Strout et al., 2019) and sparsity regularization (Zhang et al., 2019). Unlike them, we never introduce any external knowledge but highlight the inputs whose removal would significantly decrease Transformer's performance. Another work line aims to make attention better indicative of the inputs' importance (Kitada and Iyatomi, 2020; Tutek and Snajder, 2020; Mohankumar et al., 2020) which is designed for analysis with no significant performance gain, while our methods incorporate the analytical results to enhance the NMT performance.

7 Conclusion

In this paper, we present a mask perturbation model to automatically discover the decisive inputs for the model prediction. We propose three methods to calibrate the attention mechanism by focusing on the discovered vital inputs. Extensive experimental results show that our approaches obtain significant improvements over the state-of-the-art system. Analytical results indicate that our proposed methods make the low layer's attention weights more dispersed to grasp multiple information. In contrast, high-layer attention weights become more

focused on specific essential inputs. We further find a greater need for calibration in the original attention weights with high entropy. Our work provides insights on future work about learning more useful information via attention mechanisms in other attention-based frameworks.

Acknowledgments

The research work has been funded by the Natural Science Foundation of China under Grant No. U1836221 and the National Key Research and Development Program of China under Grant No. 2018YFC0823404. The research work in this paper has also been supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0504). This work is also supported by Youth Innovation Promotion Association CAS No. 2017172.

References

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

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

Yonatan Belinkov and James R. Glass. 2019. Analysis methods in neural language processing: A survey. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3348–3354. Association for Computational Linguistics.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *CoRR*, abs/1607.01628.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 876–885. The Association for Computational Linguistics.

Mengnan Du, Ninghao Liu, and Xia Hu. 2020. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77.

Yang Feng, Wanying Xie, Shuhao Gu, Chenze Shao, Wen Zhang, Zhengxin Yang, and Dong Yu. 2020. Modeling fluency and faithfulness for diverse neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 59–66. AAAI Press.

Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4452–4461. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1243–1252.

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, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (un)reliability of saliency methods. In Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 267–280. Springer.

Shunsuke Kitada and Hitoshi Iyatomi. 2020. Attention meets perturbations: Robust and interpretable attention with adversarial training. *CoRR*, abs/2009.12064.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open

source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102. The COLING 2016 Organizing Committee.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. Association for Computational Linguistics.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288. Association for Computational Linguistics.

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Thuong-Hai Pham, Dominik Machácek, and Ondrej Bojar. 2019. Promoting the knowledge of source syntax in transformer NMT is not needed. *Computación y Sistemas*, 23(3).

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. Association for Computational Linguistics.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2662–2670.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

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

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Julia Strout, Ye Zhang, and Raymond Mooney. 2019. Do human rationales improve machine explanations? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–62. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Martin Tutek and Jan Snajder. 2020. Staying true to your word: (how) can attention become explanation? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 131–142. Association for Computational Linguistics.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across NLP tasks. *CoRR*, abs/1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates Inc.

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

Elena Voita, Rico Sennrich, and Ivan Titov. 2020. Analyzing the source and target contributions to predictions in neural machine translation. *CoRR*, abs/2010.10907.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5797–5808. Association for Computational Linguistics.

Rongxiang Weng, Heng Yu, Xiangpeng Wei, and Weihua Luo. 2020. Towards enhancing faithfulness for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2675–2684, Online. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jiajun Zhang, Yang Zhao, Haoran Li, and Chengqing Zong. 2019. Attention with sparsity regularization for neural machine translation and summarization. *IEEE ACM Trans. Audio Speech Lang. Process.*, 27(3):507–518.