Enlivening Redundant Heads in Multi-head Self-attention for Machine Translation

Tianfu Zhang[†‡], **Heyan Huang**[†‡], **Chong Feng**[†♣*], **Longbing Cao**[♠]
[†]Beijing Institute of Technology
[‡]Intelligent Information Processing and Contents Computing, Key Laboratory of MIIT
[♣]The Southeast Information Technology Research institute of BIT
[♠]University of Technology Sydney
{tianfuzhang,hhy63,fengchong}@bit.edu.cn
LongBing.Cao@uts.edu.au

Abstract

Multi-head self-attention recently attracts enormous interest owing to its specialized functions, significant parallelizable computation, and flexible extensibility. However, very recent empirical studies show that some self-attention heads make little contribution and can be pruned as redundant heads. This work takes a novel perspective of identifying and then vitalizing redundant heads. We propose a redundant head enlivening (RHE) method to precisely identify redundant heads, and then vitalize their potential by learning syntactic relations and prior knowledge in text without sacrificing the roles of important heads. Two novel syntax-enhanced attention (SEA) mechanisms: a dependency mask bias and a relative local-phrasal position bias, are introduced to revise self-attention distributions for syntactic enhancement in machine translation. The importance of individual heads is dynamically evaluated during the redundant heads identification, on which we apply SEA to vitalize redundant heads while maintaining the strength of important heads. Experimental results on WMT14 and WMT16 English→German and English→Czech language machine translation validate the RHE effectiveness.

1 Introduction

Recently, self-attention network (SAN) (Lin et al., 2017) has been applied to various natural language processing tasks. Instead of drawing distance-aware dependencies like recurrent neural network (Hochreiter and Schmidhuber, 1997) and convolutional neural network (Kim, 2014), SAN captures short- and long-range relations between elements. SAN involves all signals with a weighted averaging operation, which may incorporate too many unrelated elements to concentrates on specific relations. Recent work has modified SAN to enhance specific relation learning. For example, in (Shen et al., 2018), a directional self-attention network (DiSAN) uses one to multiple positional masks to model the asymmetric attention between two elements and capture context-aware relations for all tokens. (Yang et al., 2018) modeled the local information by revising the attention distribution with a learnable Gaussian bias to focus on neighboring relations. (Shaw et al., 2018) extended SAN to efficiently consider distinct representations of the relative linear position relations between sequence elements. However, the above approaches consider the multi-head SAN as a whole but ignore unbalanced contribution distributions between heads.

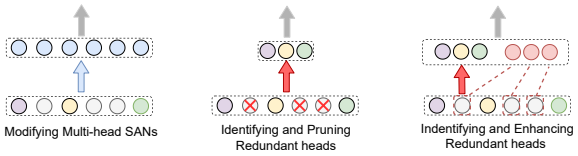


Figure 1: Rationale of the multi-head SAN. The left and middle parts are two existing SAN methods, the right one illustrates our proposed method. The colored circles represent different functions of individual heads.

Furthermore, multi-head SAN combines different attentions from multiple subspaces to construct Transformer (Vaswani et al., 2017) and achieves the state-of-the-art results in recent neural machine translation (NMT) tasks (Hassan et al., 2018). The very recent work (Voita et al., 2019) shows that the encoder-side individual heads in Transformer make different contributions, multi-heads can be classified into important heads and redundant heads and pruning redundant heads does not seriously affect performance. They also assume that important heads play various roles which influence the generated translations to different extents, including syntactic function (focusing on dependent relations), positional function (focusing on neighboring words), and rare words-based function.

To date, our understanding of the roles of dis-

*Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3238–3248
November 7–11, 2021.

tinct multi-heads is very limited, with no systematic analysis available of the roles of different heads. In this paper, we precisely identify redundant heads at the encoder-side of Transformer and demonstrate the potential of syntactically reactivating the redundant heads to improve the multi-head SAN performance. Fig. 1 illustrates the different rationales of existing work against ours in multi-head SAN. The left part represents those approaches that directly enhance overall heads as a whole w.r.t. their designed functions but do not differentiate their roles. Such approaches may downplay the functions of important heads and the diversity of the multi-head mechanism. The middle part represents the methods that analyze contributions and functions of multi-head SAN and then prune the determined redundant heads but rely on those important heads only. As shown in the right part, this paper proposes a dynamic and unified strategy to identify redundant heads and then enliven them to fulfill their potential. By enlivening the redundant heads, our approach enhances the performance of redundant heads without sacrificing the essential functions of important heads. In addition, our method further increases the scale of important heads.

Specifically, we take NMT as an example to illustrate our method of identifying and reactivating the redundant heads in multi-head SAN. We firstly propose two novel Syntax-Enhanced Attention (SEA) mechanisms for machine translation: 1) the Dependency-Enhanced Attention to use a dependent matrix as mask to model the intensive attention between dependent elements and filter elements without direct dependent relations; and 2) Local-phrase-Enhanced Attention to incorporate a distinct and learnable relative local-phrasal position matrix as bias, which is transformed from a constituency tree under the rules of *local-phrase*. These syntax-enhanced attention mechanisms simulate the specific functions of important heads but differ from the existing self-attention improvement approaches. Compared to the dependency tree, there is distinct syntactic layer information for each word in the constituency tree, which is extracted to calculate the relative phrasal position to reflect syntactic relations between elements. To this end, we define a novel phrase type *local-phrase* to only extract syntactically related words as phrase by leveraging the constituency tree, regardless of sequence distance. Further, we propose a dynamic and lightweight Redundant Heads Enlivening (RHE) strategy for multi-head SAN to reactivate and enhance the roles of redundant heads. Lastly, a dynamic function gate is designed, which is transformed from the average of maximum attention weights to compare with syntactic attention weights and identify redundant heads which do not capture meaningful syntactic relations in the sequence.

We test the above design on three widely-used translation tasks WMT14 and WMT16 English→German and WMT16 English→Czech. Extensive analyses reveal that enlivening redundant heads in multi-head SAN beats improving overall heads, and the proposed syntax-enhanced attention mechanisms with dependency and local phrases further effectively improve the translation performance.

2 Related Work

One popular extension to the SAN is to revise attention distribution by static and dynamic biases. Different dimensions of biases have been considered, including directional relation (Shen et al., 2018) and localness (Sperber et al., 2018; Zhang et al., 2018a; Yang et al., 2018). (Shen et al., 2018) improves SAN with directional masks and multi-dimensional features by explicitly revising attention distribution. In this paper, we focus on the explicit syntactic biases by proposing dependency-enhanced attention and local-phrase-enhanced attention. Several papers show that explicitly modeling dependency (Bastings et al., 2017; Nadejde et al., 2017) or phrase (Wang et al., 2017; Huang et al., 2018; Zhang et al., 2018b, 2020) is useful for tasks such as NMT. Related to our work, (Strubell et al., 2018) and (Hao et al., 2019) also modify parts of self-attention heads with syntactic information. However, they randomly assign heads instead of analysing the importance and function of each head in advance. (Sperber et al., 2018) restricts SAN with the neighboring elements and performs better for longer sequences in acoustic modeling and natural language inference tasks. (Yang et al., 2018) leverages Gaussian bias predicted by the query vector to dynamically model the localness for SAN.

Other work analyzes the attention weights of different NMT models (Ghader and Monz, 2017; Voita et al., 2018; Tang et al., 2018; Raganato and Tiedemann, 2018). (Voita et al., 2019) considers how different heads correspond to specific relations and proves that redundant heads can be pruned

without greatly decreasing translation performance. However, they disregard the full potential of redundant heads as in our SEA. (Li et al., 2018) realizes the diversity of multiple attention heads and introduces a disagreement regularization to explicitly encourage the diversity. Nevertheless, they do not realize that only partial individual heads are redundant, which is a prerequisite for optimizing multi-head diversity.

In summary, while some of the related work recognizes the approach of revising attention distribution with bias, our work represents the first to propose a complement and precise strategy to analyze individual heads, identify redundant heads and then enliven them with syntactic bias.

3 Background

3.0.1 Multi-head Self-attention

Multi-head SAN (Vaswani et al., 2017; Shaw et al., 2018; Shen et al., 2018; Yang et al., 2018) projects the input sequence to multiple subspaces (h attention heads), applies the scaled dot-product attention to the hidden states in each head, and then concatenates the output. For each self-attention head $\mathbf{head}_i$ $(1 \leq i \leq h)$ in the multi-head SAN for NMT, given an input sequence $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, each hidden state in the l-layer is constructed by attending to the states in the $(l-1)$-th layer. Specifically, the hidden states of $(l-1)$-th layer $\mathbf{H}^{l-1} \in \mathbf{R}^{n \times d_h}$ are firstly transformed into the queries $\mathbf{Q} \in \mathbf{R}^{n \times d_h}$, the keys $\mathbf{K} \in \mathbf{R}^{n \times d_h}$, and the values $\mathbf{V} \in \mathbf{R}^{n \times d_h}$ with three separate weight matrices, where d_h represents the dimensionality of each head.

The hidden state $\mathbf{H}_i$ of the l-th layer is calculated as:

$$\mathbf{H}_i^l = \sum_{j=1}^{n} Att(\mathbf{Q}_i, \mathbf{K}_j)(\mathbf{V}_j \mathbf{W}^V) \quad (1)$$

where $Att(\cdot)$ is a scaled dot-product attention model, defined as:

$$Att(\mathbf{Q}_i, \mathbf{K}_j) = softmax\left(\frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K)^T}{\sqrt{d_k}}\right) \quad (2)$$

where $\sqrt{d_k}$ is the scaling factor with d being the dimensionality of layer states.

3.0.2 Multi-head Analysis

In (Voita et al., 2019), a "confidence" scalar $\boldsymbol{h}_{conf}$ is calculated as the average of maximum attention weights of all n source tokens in one head:

$$\boldsymbol{h}_{conf} = \frac{1}{n} \sum_{i=1}^{n} Max\left(Att(\mathbf{Q}_i, \mathbf{K}_j)\right) \quad (3)$$

$Max\left(Att(\mathbf{Q}_i, \mathbf{K}_j)\right)$ represents the maximum attention weight to $\mathbf{x}_i$ among all source tokens $\mathbf{x}_j$ in the sequence. Further, a fixed gate value $\boldsymbol{f}_{gate}$ $(0 < \boldsymbol{f}_{gate} \leq 1)$ is given that judges a head as important if $\boldsymbol{h}_{conf} > \boldsymbol{f}_{gate}$ for all training examples and epochs. In addition, three head functions are identified according to the frequency of maximum attention weight assigned to a specific position: syntactic function, positional function, and rare words function.

4 The RHE Design

Fig. 2 shows the architecture of our proposed redundant heads enlivening (RHE) approach to identify redundant heads and then enliven them by revising self-attention distributions with a syntactic bias. RHE takes full advantage of the multi-head SAN by capturing both dependent and distinct phrasal relations. First, two Syntax-Enhanced Attention (SEA) mechanisms: Dependency Enhanced Attention (DEA) and Local-phrase Enhanced Attention (LPEA), are proposed. DEA disables the attention between elements without dependencies by leveraging the dependency mask, and LPEA precisely regulates the self-attention distribution by a distinct and learnable *local-phrase* bias. The bias represents relative local-phrasal position transformed from a constituency tree. LPEA precisely captures both short- and long-term syntactic relations. Second, the Redundant Head Identification module dynamically determines the importance and function of each head during the training process per the average sum of *syntactic attention weights*. Lastly, the self-attention of redundant heads is replaced by SEA to enliven their full potential and roles.

4.1 SEA: Syntax-Enhanced Attention

4.1.1 DEA: Dependency-Enhanced Attention

DEA is a syntactic extension of standard self-attention. DEA focuses on the internal dependency between elements. We place a dependency mask bias $\boldsymbol{d}$ to the logit similarity in Eq. (2):

$$Att(\mathbf{Q}_i, \mathbf{K}_j) = softmax\left(\frac{(\mathbf{x}_i \mathbf{W}^{\mathbf{Q}})(\mathbf{x}_j \mathbf{W}^{\mathbf{K}})^T}{\sqrt{d_k}} + \mathbf{D}_{i,j}\mathbf{1}\right) \quad (4)$$

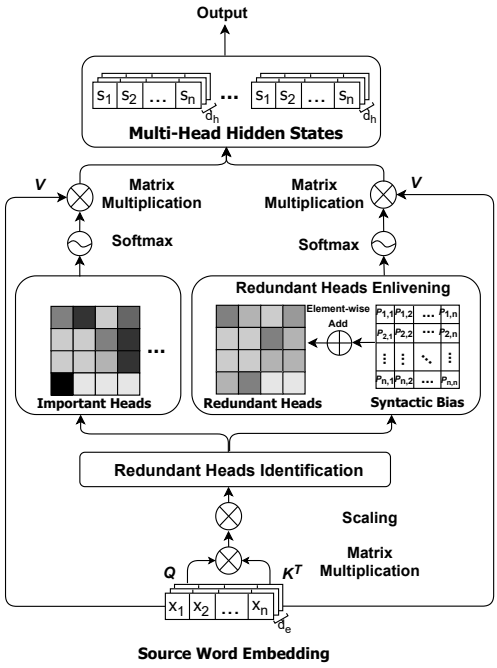


Figure 2: The architecture of our Redundant Head Enlivening model. The darker colored blocks in individual heads represent higher attention weights of the current element.

Given a dependency mask $\mathbf{D} \in \{0, -\infty\}^{n \times n}$, we set the bias $\boldsymbol{d}$ to a constant vector $\mathbf{D}_{i,j}\mathbf{1}$ in Eq. (4), where $\mathbf{1}$ is an all-one vector. Note that, due to the exponential operation in the softmax function, adding the alignment score with a bias $\boldsymbol{d} \in \{0, -\infty\}^{n \times n}$ approximates to multiplying the attention distribution by a weight $\in [1, 0)$.

To encode the dependency information into this mask, we define the value of $\mathbf{D}_{i,j}$ according to head-dependent relations $Dep\ (\mathbf{x}_i, \mathbf{x}_j)$ between elements $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\mathbf{D}_{i,j} = \begin{cases} 0, & \mathbf{x}_i,\ \mathbf{x}_j\ in\ Dep\ (\mathbf{x}_i, \mathbf{x}_j)\ or\ i = j \\ -\infty, & \mathbf{x}_i,\ \mathbf{x}_j\ not\ in\ Dep\ (\mathbf{x}_i, \mathbf{x}_j) \end{cases} \tag{5}$$

In fact, Eq. (5) shows that we ignore the relations between independent word pairs $(\mathbf{x}_i, \mathbf{x}_j)$ by set $\mathbf{D}_{i,j} = -\infty$; meanwhile, the attention weights are more concentrated on dependent word pairs. By assuming each dependent relation to be equally important, we do not assign different biases for different dependency word pairs by set $\mathbf{D}_{i,j} = 0$. This enhances the ability of self-attention to capture dependent relations.

4.1.2 LPEA: Local-phrase-Enhanced Attention

LPEA includes a distinct and learnable syntactic bias to revise the attention weights. A local-phrase bias $\boldsymbol{p}$ represents relative phrasal position information between $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{x}_j \in local\_phrase(\mathbf{x}_i)$). Meanwhile, it masks the attention between words not in $local\_phrase(\mathbf{x}_i)$. Similar to DEA, we modify Eq. (2) as:

$$Att(\mathbf{Q}_i, \mathbf{K}_j) = softmax\left(\frac{(\mathbf{x}_i\mathbf{W}^{\mathbf{Q}})(\mathbf{x}_j\mathbf{W}^{\mathbf{K}})^T}{\sqrt{d_k}} + \mathbf{P}_{i,j}\right) \tag{6}$$

We further introduce the concept of *local-phrase* obtained from the constituency tree in terms of two rules, different from general phrases which mostly consist of neighboring words. A local-phrase contains syntactically related words regardless of sequence distance, hence local-phrase carries the distinct and hierarchical syntactic relations between elements.

- **Rule 1:** Given a constituency tree with m layers, the word $\mathbf{x}_i$ and its ancestor node sequence $ast = (ast_{layer(\mathbf{x}_i)-1}, ..., ast_0)$, we assume that its $local\_phrase(\mathbf{x}_i)$ contains words which belong to the lowest multi-descendant ancestor $ast_{layer(\mathbf{x}_i)-m}$ $(0 \leq m \leq layer(\mathbf{x}_i))$.
- **Rule 2:** If word $\mathbf{x}_i \in local\_phrase(\mathbf{x}_j)$ $(j < i)$ according to **Rule 1**, we assume that word $\mathbf{x}_j \in local\_phrase(\mathbf{x}_i)$.

To obtain the local-phrase bias $\boldsymbol{p}$, we firstly extract a relative phrasal position matrix $\mathbf{RP}$ from the constituency tree.

As Fig.3 shows, first, given a matrix of $\mathbf{RP} \in \mathbf{R}^{n \times n}$, where each element represents the relative syntactic distance between words x_i and x_j. Then, for words x_i and x_j not in the same local-phrase (e.g. *"Sharon"* and *"talk"*), we set the relative position as ∞ (3^{th} row, 6^{th} column). Finally, for words which in a local-phrase, such as "held" and "talk", we calculate the relative phrasal position distance according to their relative phrase layer ($Layer_3 - Layer_4 = -1$) and set the $\mathbf{RP}_{2,4} = 1$. Accordingly, we obtain the matrix $\mathbf{RP}$.

As the $\mathbf{RP}$ matrix cannot be directly encoded in attention distribution, inspired by (Shaw et al.,

2018), We use a group of vectors to represent the relative phrasal position between words in $\mathbf{RP}$.

Considering that the precise relative phrasal position information beyond a certain distance is not useful, the maximum relative phrasal position is clipped to a maximum absolute value of k. Therefore, we consider $2k+1$ unique edge labels for relative phrasal position vectors and transform the integral matrix $\mathbf{RP}$ into the corresponding vector matrix $\mathbf{M} \in \mathbf{R}^{n \times n \times d_h}$, where:

$$\begin{aligned} \mathbf{M}_{ij} &= \mathbf{w}_{clip(j-i,k)} \\ clip(x,k) &= max(-k, min(k,x)) \end{aligned} \tag{7}$$

Then, we learn the relative phrasal position representations $\mathbf{w} = (\mathbf{w}_{-k}, ..., \mathbf{w}_k)$, where $\mathbf{w}_i \in \mathbf{R}^{d_h}$. After obtaining the matrix $\mathbf{M}$, we apply a feed-forward network to transform the relative local-phrasal position vector $\mathbf{M}_{ij}$ to a relative local-phrasal position hidden state. It is further mapped to a negative scalar $\mathbf{P}_{ij}$ of local-phrase bias matrix $\boldsymbol{p}$ by a linear projection $U_{\mathbf{P}} \in \mathbf{R}^{d_h \times 1}$, namely:

$$\mathbf{P} = -|\tanh(\mathbf{W}_P \mathbf{M} + \mathbf{b}_P)\mathbf{U}_P| \tag{8}$$

$W_P \in \mathbf{R}^{d_h \times d_h}$ and $b_P \in \mathbf{R}^{d_h}$ are model parameters. Fig. 3 shows the process of extracting relative local-phrase bias $\boldsymbol{p}$ from the constituency tree.

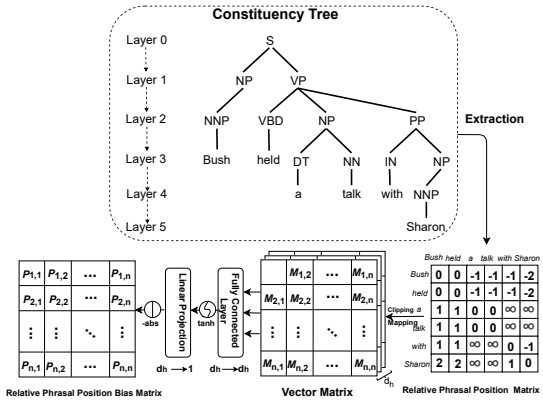


Figure 3: The process of extracting relative phrasal position bias.

4.2 Incorporating SEA into Multi-head Self-attention

4.2.1 Redundant Head Identification

We enhance the syntactic function of self-attention heads by dynamically identifying the redundant heads that lack the ability of capturing both short- and long-term syntactic relations to enhance these heads by incorporating SEA. We firstly apply the dependency mask Dep_mask to the attention weight matrix to obtain the corresponding syntactic attention weights which reflect short- and long-term syntactic relations. Then, we sum the syntactic attention weights for each $\mathbf{x}_i$ among all syntax-related source tokens $\mathbf{x}_j$ in the sequence. Finally, we calculate the average of syntactic attention weight scalar Syn_{attn} as follows:

$$Syn_{attn} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} Dep\_mask\,(Att(\mathbf{Q}_i,\ \mathbf{K}_j)) \tag{9}$$

We propose a function gating criteria: when the average of syntactic attention weights is higher than the average of maximum attention weights, the head is regarded as important and contains syntactic functions. Different from the work in (Voita et al., 2019) which simply uses a fixed gate value to measure the importance of individual head for all training examples and epochs, our method dynamically identifies individual heads for each sentence during the training process. We compare syntactic attention weights Syn_{attn} with dynamic and learnable syntactic gate Syn_{gate} transformed from head confidence $\boldsymbol{h}_{conf}$ in Eq. (3) by sigmoid activation functions, i.e., $Syn_{gate} = sigmoid(\boldsymbol{h}_{conf})$ to determine the head function. If Syn_{attn} is lower than Syn_{gate}, we treat the corresponding head as redundant.

$$\boldsymbol{h}_{label} = \begin{cases} 1, & Syn_{attn} > Syn_{gate} \\ 0, & other \end{cases} \tag{10}$$

$\boldsymbol{h}_{label}$ represents whether a head is important ($\boldsymbol{h}_{label} = 1$) or redundant ($\boldsymbol{h}_{label} = 0$). Another aspect of additional reason for comparing with the head confidence is that some

4.2.2 Enlivening Redundant Heads

After differing redundant heads from those important ones in the multi-head self-attention, we further enliven the redundant heads with a syntactic bias per Eq. (4) or Eq. (6) without interfering with the important head functions. (Voita et al., 2019) shows that redundant heads are mostly distributed in the lower encoder layers, meanwhile (Hao et al., 2019; Yang et al., 2018) shows that the bottom layer in the encoder, which directly takes word embedding as input, benefits more from modeling local relations. We evaluate the performance of applying our method on the low- and high-level encoder layers in the next section, and obtain the best

performance when applying on the first encoder layer.

5 Experiments

5.1 Settings

We carry out experiments on the English→German (En→De) and English→Czech (En→Cs) language translation. For En→De, the classic WMT14 data consists of 4.5M sentence pairs (newstest2013 and newstest2014 as development set and test set), and the WMT16 News Commentary v11 data consists of 0.22M sentence pairs (newstest2015 and newstest2016 as development and test sets). For En→Cs, the WMT16 News Commentary v11 data data consists of approximately 0.18M sentence pairs (newstest2015 and newstest2016 as development set and test set). We evaluate our approach in terms of different languages and data sizes. We use the Berkeley Neural Parser (**?**) to generate constituency trees for English, and an open-source tool spaCy[1] to parse dependency trees for English. Besides, we make statistical significance test with the method in (Collins et al., 2005). The byte-pair encoding (BPE) toolkit[2] (Sennrich et al., 2016) is used with 32K merge operations. The 4-gram NIST BLEU score (Papineni et al., 2002) is used as the evaluation metric. We implement the proposed RHE and all the baselines on top of Transformer model (Vaswani et al., 2017) by using open-source toolkit OpenNMT (Klein et al., 2017). Please refer to the Appendix for more details of dataset and parameter setting .

5.2 RHE for NMT Results

Table 1 shows the ablation study results of the Transformer enabled by the two proposed SEA mechanisms DEA and LPEA and the RHE approach.

First, the Rows of "+DEA" and "+LPE" represent the models with all heads of the first encoder layer, including original important heads, are replaced by the syntax-enhanced attention networks DEA and LPEA respectively. Second, the RHE approach (containing the Rows of "+DEA+RHE" and "+LPEA+RHE") significantly lifts both DEA and LPEA mechanisms across all small and large language pairs. This tests the effectiveness of identifying and modifying redundant heads without interfering important head functions. RHE lifts the LPEA, which together i.e. LPEA+RHE substantially outperforms Transformer by +1.0 BLEU points on En→De (WMT16), +0.96 BLEU points on En→De (WMT14), and +0.81 BLEU points on En→Cs (WMT16). These results demonstrate the efficacy and applicability of both SEA and RHE designs.

The upper part of Table 1 shows the results of Transformer enabled by two SAN enhancement strategies: the relative position encoding method (Rel_Pos) (Shaw et al., 2018) which considers the relative position between sequence elements, and the modeling localness (Localness) (Yang et al., 2018) method which enhances the ability of capturing local context for self-attention with a learnable Gaussian bias. While both Rel_Pos and Localness make improvement over Transformer owing to their strategies of enhancing SAN, our DEA, DEA+RHE, LPEA and LPEA+RHE-enabled Transformers substantially and consistently beat the standard Transformer and both Rel_Pos and Localness-enhanced Transformers. For example, our DEA+RHE on Transformer outperforms Rel_Pos by over 0.49 BLEU points on En→De (WMT16), 0.29 BLEU points on En→De (WMT14), and 0.36 BLUE points on En→Cs (WMT16). This is owing to the SEA and RHE design of assigning a distinct syntactic bias for each word and modeling both short- and long-term syntactic relations.

5.3 RHE Mechanism Analysis

Here, we analyze the RHE generalizability, the impact of different factors, and the visualization of multi-head attention matrices. Owing to space limitation, we only report the testing results on the En→De (WMT16) set, and explore the influence caused by syntax parsing quality and applied encoder layers in Appendix.

5.3.1 The RHE Applicability

Table 2 shows that RHE lifts Rel_Pos and Localness by +0.28 and +0.20 BLEU point respectively. This proves (1) RHE is general and can enhance other multi-head SAN; and (2) the necessity of preserving important heads while improving multi-head self-attention mechanisms. By pruning redundant heads, the experiment also shows that RHE can precisely identify redundant heads and the RHE-enabled Transformer only drops 0.1 BLEU point after pruning the identified redundant heads, meanwhile the training speed improves slightly.

[1] https://spacy.io

[2] https://github.com/rsennrich/subword-nmt

Architecture	En→De (WMT16)		En→De (WMT14)		En→Cs (WMT16)	
	#Para	BLEU	#Para	BLEU	#Para	BLEU
Transformer	71.82M	25.28	88.00M	27.31	70.02M	15.46
+ Rel_Pos	71.85M	25.49	88.10M	27.53	70.05M	15.60
+ Localness	71.84M	25.53	88.80M	27.61	70.04M	15.65
+ DEA	71.82M	25.75	88.10M	$27.71^{\uparrow}$	70.02M	$15.84^{\uparrow}$
+ DEA + RHE	71.82M	$25.98^{\Uparrow}$	88.10M	$27.82^{\uparrow}$	70.02M	$15.96^{\Uparrow}$
+ LPEA	71.82M	$25.90^{\Uparrow}$	88.10M	$27.96^{\uparrow}$	70.02M	$15.97^{\Uparrow}$
+ LPEA + RHE	71.82M	**26.28**$^{\Uparrow}$	88.10M	**28.27**$^{\Uparrow}$	70.02M	**16.27**$^{\Uparrow}$

Table 1: Test results of SEA and RHE against baseline SAN-enhanced Transformer for NMT on WMT16 and WMT14 En→De, and WMT16 En→Cs. "# Para" denotes the trainable parameter size of each model (M = million). Symbols "$\uparrow$ /$\Uparrow$" refer to the improvement significance level over the self-attention baseline ($p < 0.05/0.01$) tested by bootstrap resampling.

This shows the importance of precisely identifying redundant heads, and only by then pruning redundant heads would trivially affect the learning performance as shown in (Voita et al., 2019).

Systems	Speed	BLEU	△
Transformer	1.21	25.28	-
+RHE (Prune)	1.23	25.18	- 0.10
Rel_Pos	1.17	25.49	+ 0.21
+RHE	1.17	**25.77**	+ 0.49
Localness	1.18	25.53	+ 0.25
+RHE	1.17	**25.73**	+ 0.45

Table 2: Impact of RHE on two multi-head SAN methods Rel_Pos and Localness and pruning redundant heads. "Speed" denotes the training speed (steps/second).

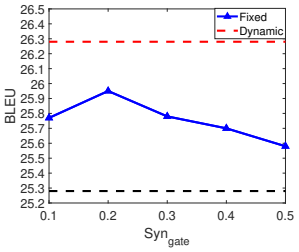


Figure 4: Comparison between fixed syntactic function gate values and dynamic syntactic function gate (the red line), and the black dashed line represents the baseline.

5.3.2 Selection of Multi-head Function Gate

Two strategies can be used to select the multi-head function gate: one is a fixed gate by a constant number throughout the whole training process; the other is a dynamic gate transformed from the average of maximum attention weight $\boldsymbol{c}$ of an individual head, which provides a flexible criteria to determine the head function.

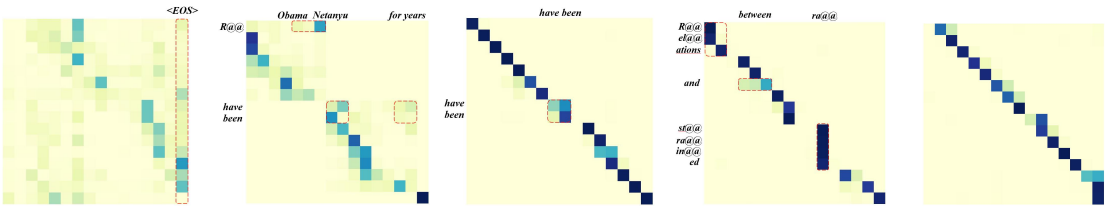
Fig. 4 shows the comparison between multiple fixed gate values and the dynamic gate. We adjust the value of the fixed gate in a range (0.1, 0.5)[3].

The results show that the dynamic gate strategy significantly outperforms all fixed gate values. The performance becomes unstable when the fixed gate value increases. Self-attention heads develop their ability to capture syntactic relations during the training epochs; accordingly, the average syntactic attention weights Syn_{attn} increase gradually. Low fixed gate value reduces the recall of RHE because Syn_{attn} goes high in later epochs; high fixed gate value reduces the accuracy of RHE as all important heads and redundant heads receive small Syn_{attn} in the initial epochs. Hence, the high fixed gate might mistakenly treat a high portion of heads as redundant.

5.3.3 Effect of Maximum Relative Local-Phrasal Position

Compared to the dependency tree, the constituency tree characterizes the distinct relative phrasal position for each word, which enriches the syntactic relations between elements. We thus evaluate the effect of varying the clipping distance k of the maximum absolute relative local-phrasal position. The results in Table 3 show that the performance increases with the increase of k from 0 to 6, while

[3]Once the average of syntactic attention weights satisfies $Syn_{attn} > 0.5$, it is higher than the average non-syntactic attention weights, hence we assume that the head is functional.



(a) Redundant Head (b) LPEA+RHI Head (c) Rel_Pos Head (d) Localness Head (e) Important Head

Figure 5: Visualization of attention matrices of the same input sentence and the same encoder layer.The darker color of a cell represents higher attention weight of the source token.

this trend does not hold when $k = 8$.

The average of maximum phrase layers of the training set is 11.13, which is close to the maximum absolute relative phrasal position $k = 5$ and $k = 6$ (where $2k + 1$ is 11 and 13). This result indicates that the best performance appears when the relative phrasal position vector exactly covers the average of the maximum phrase layer.

k	0	2	4	6	8
BLEU	25.28	25.57	25.86	**26.28**	25.51
△	-	+0.29	+0.58	**+1.0**	+0.23

Table 3: Results w.r.t. the clipping relative local-phrase layer distance k.

5.3.4 Visualization of LPEA+RHE-enlivened Attention

To evaluate the effect of LPEA+RHE-enlivened redundant heads against Rel_Pos and Localness, we further visualize the attention matrices of an individual head in the first encoder layer. The source sentence is *Relations between Obama and Netanyhu have been strained for years* $\langle EOS \rangle$.

The improvement between redundant head and LPEA+RHE-enlivened head is shown in Fig. 5 (a) and (b). In Fig. 5 (a), the distribution of original redundant head attention concentrates more on the end of the sentence (16^{th} column) but less on the specific meaningful words. In Fig. 5 (b), SEA masks those words that do not belong to the local-phrase in each row and improves the attention in local-phrase: 1) 'have been ... for years' in rows 8 and 9, which is a long-distance and discontinuous phrase; 2) SEA strengthens the attention between 'Relations' and 'Obama', 'Netanyhu' in the 1^{st} row, which has the *nmod* dependency.

Fig. 5 (c) and (d) shows the results of Rel_Pos and Localness, both explicitly models the locality for self-attention networks. Both of their attention weights mainly distribute along the diagonal and some short-range elements. Rel_Pos captures the phrase 'have been' in rows 8 and 9 but ignores long-range phrase elements 'for years' since the influence of relative position representation decays as the sequence distance increases. In Fig. 5 (d), the attention weight distribution of Localness is more flexible because they assign a distinct Gaussian bias to each position, which pays more attention to the local syntactic context. It captures the phrase 'between...and' in the 6^{th} row. However, the attention may focus on the word itself sometimes, such as the high attention weights of 'Relations' ('R el ations' in the subword form) in the 1^{st} column and 'strained' ('st ra in ed' in the subword form) in the 11^{th} column. In contrast, LPEA+RHE enlivens the redundant head by modeling the latent syntactic localness beyond the constraints of sequence distance. Fig. 5 (e) shows the attention matrix of an important head, which focuses on neighboring words. This result is consistent with the previous findings in (Voita et al., 2019).

6 Conclusions

While multi-head self-attention networks show a significant potential in improving learning tasks such as NMT, an open challenging topic is to quantify the redundancy and importance of each head and further improve the weak heads. This paper makes one step forward by not only precisely analyzing and identifying redundant heads but introducing a dynamic redundant heads enlivening (RHE) mechanism to identify and enliven each redundant head toward full potential without affecting the function of other important heads as in alternatively enhancing all heads. The proposed dependency-enhanced attention and local-phrase-enhanced attention effectively capture the different syntactic relations between elements. We'll work on strategies to integrate DEA and LPEA in future.

7 Acknowledgements

We very appreciate the comments from anonymous reviewers which will help further improve our work. This work is supported by National Key R&D Plan (No.2018YFC0832104), National Natural Science Foundation of China (No.61732005). This work is finished during Tianfu Zhang's visit to the UTS Australia

References

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1957–1967. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *ACL*, pages 531–540.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *IJCNLP*, pages 30–39.

Jie Hao, Xing Wang, and Shuming Shi et al. 2019. Multi-granularity self-attention for neural machine translation. In *EMNLP*, pages 887–897.

Hany Hassan, Anthony Aue, and Chang Chen et al. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Po-Sen Huang, Chong Wang, and Sitao Huang et al. 2018. Towards neural phrase-based machine translation. In *ICLR*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In ACL.

Guillaume Klein, Yoon Kim, and Yuntian Deng et al. 2017. Opennmt: Open-source toolkit for neural machine translation. In *ACL*, pages 67–72.

Jian Li, Zhaopeng Tu, and Baosong Yang et al. 2018. Multi-head attention with disagreement regularization. In *EMNLP*, pages 2897–2903.

Zhouhan Lin, Minwei Feng, and Cícero Nogueira dos Santos et al. 2017. A structured self-attentive sentence embedding. In *ICLR*.

Maria Nadejde, Siva Reddy, and Rico Sennrich et al. 2017. Syntax-aware neural machine translation using CCG. *CoRR*, abs/1702.01147.

Kishore Papineni, Salim Roukos, and Todd Ward et al. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *NAACL*, pages 404–411.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *EMNLP*, pages 287–297.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL*, pages 464–468.

Tao Shen, Tianyi Zhou, and Guodong Long et al. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *AAAI*, pages 5446–5455.

Matthias Sperber, Jan Niehues, and Graham Neubig et al. 2018. Self-attentional acoustic models. In *Interspeech*, pages 3723–3727.

Emma Strubell, Patrick Verga, and Daniel Andor et al. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*, pages 5027–5038.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *WMT*, pages 26–35.

Ashish Vaswani, Noam Shazeer, and Niki Parmar et al. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, and Rico Sennrich et al. 2018. Context-aware neural machine translation learns anaphora resolution. In *ACL*, pages 1264–1274.

Elena Voita, David Talbot, and Fedor Moiseev et al. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL*, pages 5797–5808.

Xing Wang, Zhaopeng Tu, and Deyi Xiong et al. 2017. Translating phrases in neural machine translation. In *EMNLP*, pages 1421–1431.

Baosong Yang, Zhaopeng Tu, and Derek F. Wong et al. 2018. Modeling localness for self-attention networks. In *EMNLP*, pages 4449–4458.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018a. Accelerating neural transformer via an average attention network. In *ACL*, pages 1789–1798.

Jingyi Zhang, Masao Utiyama, and Eiichiro Sumita et al. 2018b. Guiding neural machine translation with retrieved translation pieces. In *NAACL*, pages 1325–1335.

Tianfu Zhang, Heyan Huang, Chong Feng, and Xiaochi Wei. 2020. Similarity-aware neural machine translation: reducing human translator efforts by leveraging high-potential sentences with translation memory. *Neural Computing and Applications*, pages 1–13.

A Appendix

A.1 Setting Details

Our experiment dataset statistics are summarized in Table A1.

Dataset	Train	Val.	Test
En-De (WMT16)	226822	2168	2999
En-De (WMT14)	1945614	2168	2999
En-Cz (WMT16)	181112	2656	2999

Table A1: The number of train set, development set and test set sentences of three experiment datasets.

We follow the Transformer (base model) setting in (Vaswani et al., 2017) to train the models and reproduce their reported results on the En→De task. The hidden size is 512, filter size is 2,048, and the number of attention heads is 8. All models are trained on four NVIDIA TITAN Xp GPUs where each is allocated with a batch size of 4,096 tokens. We average the last 10 checkpoint models to ensure the robustness of translation performance.

A.2 Effect of Enhancing Different Layers in Encoder

The work in (Voita et al., 2019) shows that there is only one important head associated with rare words function on the first layer, while more heads are with positional and syntactic functions on higher layers. Their work indicates the necessity of lifting individual heads rather than treating them same. In this experiment, we test this by applying the local-phrase-enhanced attention to different combinations of layers in the encoder.

As shown in Table A2, enhancing the syntactic function on the first layer outperforms applying it to any other layer combinations for translation and achieves the fastest training speed due to only modifying one layer; and the performance drops with the increase of layers from bottom to top (Rows 2-5 in the table). However, enhancing the syntactic function on the higher three layers and the overall layers (Rows 6 and 1) decreases the translation performance. These results reveal that lower layers may have fewer important heads to be enhanced, while higher layers may have too many important heads, leading to harder differentiation in the enhancement. In addition, our results are consistent with the analysis in the related work (Yang et al., 2018) and (Hao et al., 2019), which shows that the lower encoder layers benefit more from modeling the localness and phrase structure. Accordingly, we only enhance the first layer of SAN in the following experiments.

#	Layers	Speed	BLEU	△
0	[0-0]	1.21	25.28	-
1	[1-6]	1.10	24.68	- 0.60
2	[1-1]	1.18	**26.28**	+ 1.0
3	[1-2]	1.17	25.93	+ 0.65
4	[1-3]	1.15	25.66	+ 0.38
5	[1-4]	1.15	25.43	+ 0.15
6	[4-6]	1.16	24.96	- 0.32

Table A2: Effect of enhancing different layers of the encoder by the local-phrase-enhanced attention without enlivening redundant heads.

A.3 Effect of Syntax Parsing Quality

We use an external constituency tree parser to generate the syntactic structure for the source sentence. Based on that, we can extract the local-phrase and characterize the relative local-phrasal position features to modify the self-attention network. Hence, the impact of the quality of different parsers on translation performance is necessary to be analysed.

We compare the effect of two classical constituency tree parser tools, PCFGs-based Parser (Petrov and Klein, 2007) and Neural-based Parser (Kitaev and Klein, 2018), on the performance of the LPEA+RHE mechanism. Table A3 shows the reported parsing performance (F1 score) on the Penn Treebank WSJ test set (for English) and its corresponding translation BLEU score in this work.

The results indicate that, the higher quality of parsing trees, the better performance of the syntax-enhanced NMT model across dataset sizes and lan-

guages, with about 0.30 BLEU points improvement. We think that the improvement of parsing and translation is owing to that the neural-based parser leverages Transformer as encoder to represent the sentence. Although exploring the best performance of parsing tools is not the focus of this work, we believe that, with higher quality of parsing tool, our SEA mechanisms have more potential to represent the syntactic bias for self-attention network.

Metric	Task	PCFG	Neural
F1	WSJ Parsing	91.20	93.55
BLEU	En-De (WMT16)	25.98	26.28
	En-De (WMT14)	28.02	28.27
	En-Cs (WMT16)	15.92	16.27

Table A3: Performance of two classical constituency tree parser tools on the Penn Treebank WSJ test set (F1 score) and its corresponding effect on the LPEA+RHE NMT model (BLEU score).