Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Enhancing low-resource neural machine translation with syntax-graph guided self-attention

Longchao Gong, Yan Li, Junjun Guo [*], Zhengtao Yu, Shengxiang Gao

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650500, China
Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, 650500, China

ARTICLE INFO

Article history:
Received 27 July 2021
Received in revised form 20 February 2022
Accepted 16 March 2022
Available online 28 March 2022

Keywords:
Neural machine translation
Low-resources
Prior knowledge incorporating

ABSTRACT

Most neural machine translation (NMT) models only rely on parallel sentence pairs, while the performance drops sharply in low-resource cases, as the models fail to mine the linguistry of the corpus. Incorporating prior monolingual knowledge explicitly, such as syntax, has been shown to be effective for NMT, particularly in low-resource scenarios. However, existing approaches have not exploited the full potential of the NMT architectures. In this paper, we present syntax-graph guided self-attention (SGSA): a neural network model that combines the source-side syntactic knowledge with multi-head self-attention. We introduce an additional syntax-aware localness modeling as a bias, which indicates that the syntactically relevant parts need to be paid more attention to. The bias is then incorporated into the original attention distribution to form a revised distribution. Moreover, to maintain the strength of capturing the meaningful semantic representations of source-sentence, we adopt a node random dropping strategy in multi-head self-attention subnetworks. Extensive experiments on several standard small-scale datasets demonstrate that SGSA can significantly improve the performance of Transformer-based NMT, and is also superior to the previous syntax-dependent state-of-the-art.

1. Introduction

End-to-end NMT, which leverages a sequential encoder–decoder network to transform a source sentence into a target one, has led to state-of-the-art performance [1,2]. Research in NMT mostly acquiesces that a model can automatically learn prior linguistic knowledge in the presence of large-scale parallel sentence pairs. However, [3,4] pointed out that traditional NMT models still fail to dig these deep structural details. On the other hand, despite the remarkable success, the quality of translation rapidly deteriorates as the amount of parallel data decreases [5]. For most language pairs, collecting large-scale high-quality bilingual sentence pairs is high time- and labor-cost, NMT still suffers from the data scarcity problem. We argue that NMT modeling methods are chaotic and need prior linguistry as explicit constraints and guidance, especially in low-resource settings.

Extracting and incorporating prior monolingual knowledge, such as syntax information, which is an obvious prerequisite for semantic parsing, has been widely used in downstream natural language processing (NLP) tasks [6–8]. As the instance shown in Fig. 1, the words "eats" and "banana" are dependent on each other with "obj" relation, this word-to-word correlation is syntactic dependency, which is a kind of predefined rule. In intuition, these predefined linguistic rules are a crucial basis for constructing semantics and parsing language [9–11].

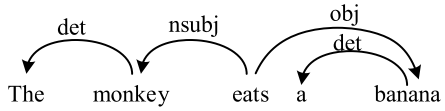


Fig. 1. Syntactic dependency for sentence "The monkey eats a banana".

Moreover, several works have shown the benefits of introducing syntactic information into NMT, especially in low-resource scenarios. For example, syntax-aware systems have shown promising BLEU improvements by either sharing models [12] or concatenating representations [13]. The new workhorse self-attention has also been optimized by incorporating explicit syntactic information [14]. Due to the introduction of extra computing components and cost, these works indicate that though enriching NMT with syntactic information has the potential to improve performance, there is not yet an architecture that maximizes the benefits of this auxiliary information.

[*] Corresponding author at: Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, 650500, China.
E-mail addresses: glc@stu.kust.edu.cn (L. Gong), 20170039@kust.edu.cn (J. Guo).

https://doi.org/10.1016/j.knosys.2022.108615
0950-7051/© 2022 Elsevier B.V. All rights reserved.

In response, we propose *syntax-graph guided self-attention* (SGSA), a model that combines source sentences syntax with stacked layers of multi-head self-attention, aiming at improving Transformer-based NMT by utilizing syntax in a more direct way. We leverage an inflectional variant of multi-head self-attention to explicitly exploit source-side syntactic dependency as a guide of the self-attention, rather than serve as an input feature. Further, to eliminate ambiguity and enhance the robustness of the model, we introduce multiple attention representations adaptively by a node random drop mechanism. Experiment results on standard En↔De, En→Tr and En→Vi translation tasks are given to show the effectiveness of our approach. The syntax combination mechanism exhibits significant improvements for low-resource NMT, while SGSA consistently outperforms the syntax-dependent state-of-the-art.

Our main contributions in this paper can be summarized as follows:

- We propose a novel, parameter-free syntactically directed self-attention mechanism that exploits the source-side syntactic dependency as a guidence, letting the model focus on the dependency parent of each token when encoding the source sentence, which effectively integrates prior syntactic knowledge without any extra cost.
- We convert the syntax into a graph so that realize an efficient combination with the NMT model. The syntax-aware approach is suitable for sub-word units, which alleviates the problems of large dictionaries and sparse words.
- We introduce multiple attention representations for stronger model robustness. We adjust the syntactic graph adaptively to realize *dynamic multiple syntax-aware self-attention representations* (DMSRs), and explore the influence of different fusion methods on the performance of the model.
- Experimental results on several standard IWSLT and WMT translation tasks demonstrate that our proposed approach achieves remarkable improvements over the state-of-the-art methods.

2. Background

As a sequence-based model, Transformer [15] is constructed by stacked layers and has emerged as the actual standard form of NLP tasks. We start with outlining the structure of the Transformer in the order of execution, including Embedding, Multi-Head Self-attention, Feed-Forward Network, and Layer-Normalization.

Embedding. Given an input sentence $x = x_1, \dots, x_n$, the Transformer encoder first converts it into an embedding vector $E_x \in R^{n\times d}$. In order to make up for the lack of position information caused by nonlinear modeling, besides word embedding, position embedding with relative position representation is injected into embedding as an essential part.

Multi-head self-attention. The input x is linearly transformed to three different matrices, i.e., query Q, key K and value V, then the dot products of attention is computed as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}}V), \tag{1}$$

where the result of QK^T is divided by $\sqrt{d}$ to make the gradient more stable, alleviate gradient disappearance that may be caused by too large dot product value. For the sake of diversified representations, the attention process is executed H times in parallel, that is, the number of multiple heads. The linear weights Q, K and V in each attention channel are different. The H sub-values are concatenated and projected again to produce the final value.

Feed-forward network. Feed-Forward Network is used for linear characterization and nonlinear transformation:

$$FFN(x) = \psi(xW_1 + b_1)W_2 + b_2, \tag{2}$$

where $\psi(\cdot)$ is the activation function such as ReLU.

Layer-normalization. Another key technique, Layer-Normalization, is used to ensure the stability of data feature distribution and accelerate the convergence speed of the model:

$$LN(x) = \alpha \times \frac{x - \mu}{\sqrt{\sigma^2}} + \beta, \tag{3}$$

where σ and μ are variance and mean respectively, α and β are trainable parameters.

The above components are iterated between layers except the embedding in the first layer, and the input of the current layer is the output of the previous layer. Furthermore, to avoid gradient disappearance and gradient explosion, which are traps that should not fall prey to, residual networks are used in each sublayer.

3. Model

In this section, we propose a syntax-graph guided self-attention model that makes effective use of the information on source sentences syntax. We introduce a variant of multi-head self-attention to the Transformer encoder. Specifically, to alleviate the lack of syntactic information and maintain the flexibility on parallel computation for self-attention networks, we represent the syntactic dependencies of each source token as vectors and apply them to the self-attention component query and key.

Fig. 2 shows the overall architecture of SGSA, the main upgrade is in the encoder, which consists of (1) a graph conversion mechanism that transforms the source sentence syntactic parsing result into a graph; (2) a syntax-aware self-attention that lever a syntax-aware mask mechanism, which apply the graph to modify self-attention syntactically; (3) a node dropping mechanism that randomly drops the graph nodes to generate multiple attention representations; (4) a fusion module to fuse the multiple representations for the follow-up tasks.

3.1. Source syntax graph

To improve the translation with syntactic knowledge, we incorporate the source-side syntactic dependency into the encoder as additional information. Whether for training or prediction, for any given source language sentence $x = x_1, \dots, x_n$, we first use an external parser to parse the syntactic relationship between words as instanced in Fig. 1, then construct token-to-token relations according to the syntactic relationship parsed by the external dependency parsing tool. In a more intuitive way, the parsing result is converted into a graph, in which the nodes represent tokens in the sentence and the edges represent their relationships. We consider the bidirectional relationship for every two related tokens, moreover, self-loop edges are added to ensure each node representation directly affects its new representation. These graphs, will eventually be provided to self-attention in the form of matrices for the sake of facilitating integration.

Since NMT struggles with out-of-vocabulary words, we construct sub-word-level syntax graphs besides word-level to adapt sub-word segmentation (e.g., Byte Pair Encoding [16]). To this end, the results of the word-level syntactic parsing are taken into account, and the sub-word connection is considered. We set the segmented sub-words as pairwise correlation, and define the sub-word relationship as a constant 1 that represents correlation in the syntactic matrices. The instance of word-level and sub-word-level syntax matrices of an input sequence "The monkey eats a banana" are shown in Fig. 3.

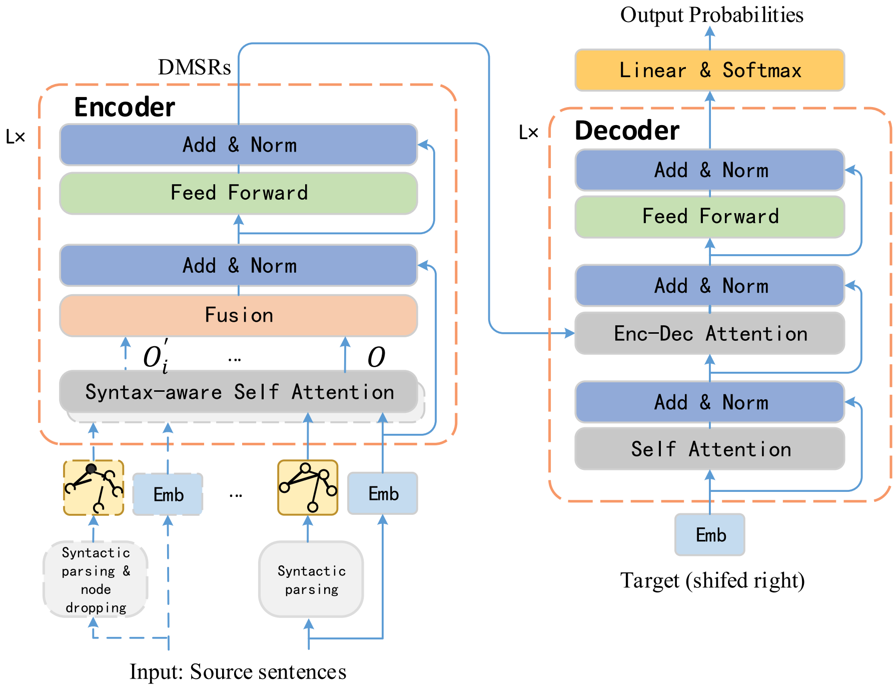
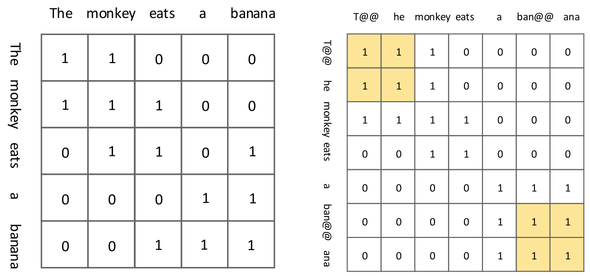


Fig. 2. Overview structure of SGSA. The dotted line portion indicates the process of generating additional multiple attention representations, using a shared syntax-aware self-attention.



	The	monkey	eats	a	banana
The	1	1	0	0	0
monkey	1	1	1	0	0
eats	0	1	1	0	1
a	0	0	0	1	1
banana	0	0	1	1	1

(a) word-level.

	T@@	he	monkey	eats	a	ban@@	ana
T@@	1	1	1	0	0	0	0
he	1	1	1	0	0	0	0
monkey	1	1	1	1	0	0	0
eats	0	0	1	1	0	0	0
a	0	0	0	0	1	1	1
ban@@	0	0	0	0	1	1	1
ana	0	0	0	0	1	1	1

(b) sub-word-level.

Fig. 3. The word-level and sub-word-level syntax matrices for the input sequence “The monkey eats a banana”. Where the constant 1 indicates correlation and 0 indicates irrelevance, which are determined according to the dependency parsing results. The yellow area identifies the segmented sub-word.

3.2. Syntax-aware self-attention

Now we set out to apply the obtained syntax graphs to guide the self-attention (a core component of the new NLP workhorse Transformer). We utilize the mask mechanism, which has been proved effective in language understanding by pre-training techniques like BERT [17] and XLNet [18]. Specifically, we perform syntax-aware masking based on the syntactic matrix to extract the syntax-dependent part as a bias, and then highlight syntax when computing the attention scores.

We represent source sentences both locally and globally. In addition to the traditional fully connected multi-head self-attention, we lever syntactic knowledge to guide self-attention a prior and strengthen the expression of syntactic related tokens by masking. For a sentence of length n, the input of each attention head is a matrix $E_x \in R^{n\times d_{model}}$ obtained by embedding, it is further transformed into three matrices $Q^h \in R^{n\times d'}$, $K^h \in R^{n\times d'}$ and $V^h \in R^{n\times d'}$ by three different linear projections $W^Q \in R^{d_{model}\times d'}$, $W^K \in R^{d_{model}\times d'}$ and $W^V \in R^{d_{model}\times d'}$. Where $d' = d/H$, and H is the head number. The dot product form attention weight calculated according to Eq. (1) is expressed as:

$$S^h = Q^h K^{h^T} / \sqrt{d}. \tag{4}$$

Then we re-score $S^h \in R^{n\times n}$ by the syntactic matrix as:

$$S^{h'} = \gamma(S^h, M^h), \tag{5}$$

where the syntactic matrix $M^h \in R^{n\times n}$, $\gamma(\cdot)$ represents syntax-aware mask, in which the syntactic related parts of S^h are retained, and the irrelevant parts are discarded. Further, the syntactically constrained $S^{h'}$ is added to S^h to emphasize the syntax. Finally, we measure the weight distribution of the sentences by a *softmax* function, and apply the result to the value V^h, obtaining the representation of SGSA head h. This process is illustrated in Fig. 4.

3.3. Node random dropping

Due to the scarcity of corpora with gold-standard parses, we rely on noisy syntactic dependency. To eliminate the noise from the external parser while obtaining multiple representations of the input sentence, we further do random dropping on the syntax graph nodes. Each syntax graph node (word or sub-word) will be dropped based on a Bernoulli distribution, to some extent, which makes the model more robust. We set the parameter as 0.1, which ensures the overall integrity of the representation to a certain extent. Syntactic dependencies between the dropped node and other nodes are also be ignored, and accordingly, the corresponding elements of the syntax matrix are set to be invalid. The graph node random dropping processes are shown in Fig. 5.

As the dotted line part shown in Fig. 2, the source sentence (embedding representations in the first layer, hidden representations in high layers) and its corresponding dropped syntax graphs are fed to the shared syntax-aware self-attention sublayer to generate additional multiple attention representations.

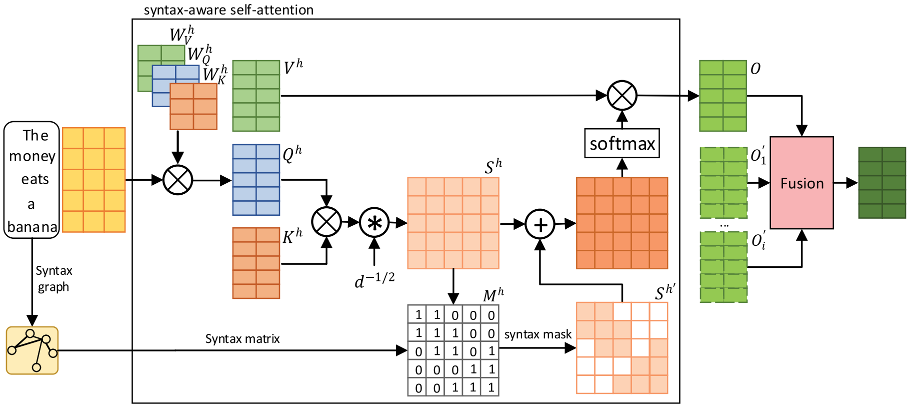
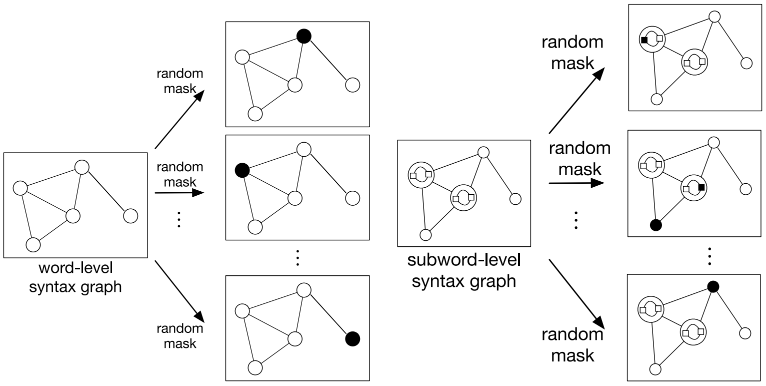


Fig. 4. The treatment of syntax-aware self-attention for the input sequence "The monkey eats a banana". The dotted part O_i' is additional multiple attention representations generated by the token random dropping mechanism. For ease of display, the multi-head fusion is not drawn in the figure.



(a) word-level.

(b) sub-word-level.

Fig. 5. Syntax graph node random dropping.

Let x and $graph$ denote the representations of the source sentence and its counterpart syntax graph respectively, and $\zeta(\cdot)$ indicate node random dropping, the above process can be summarized as:

$$O' = SSA(x, \zeta(graph)) \tag{6}$$

where $SSA(\cdot)$ is syntax-aware self-attention net described in Section 3.2. By the collaboration of syntax-aware mask and node random dropping mechanism, we realize dynamic multiple syntax-aware self-attention representations (DMSRs).

3.4. Fusion strategies

We investigate different strategies to integrate the DMSRs and get the final representation, including average pooling, highway network, and linear network. A highway network takes the following forms:

$$y = H(x, W_H) * T(x, W_T) + x * (1 - T(x, W_T)), \tag{7}$$

where x is the input and y is the output of the network. $H(x, W_H)$ and $T(x, W_T)$ are two affine functions with weight W_H and W_T respectively, which mean to perform affine transformations on x. The value output by $T(x, W_T)$ is a floating-point number belonging to (0, 1), in this sense, $T(x, W_T)$ is a gate that controls the weight of $H(\cdot)$ and x in y. In this paper, the activation functions of $H(x, W_H)$ and $T(x, W_T)$ are $Relu(\cdot)$ and $Sigmoid(\cdot)$, respectively. For highway and linear networks, we splice join each part of DMSRs as input, and dynamically adapt to the dimension of the weight matrix in affine function.

From the perspective of local attention [19], SGSA can be seen as a revision of [20]: from Gaussian distribution as the core to syntactic perception as the primary — also from model-built rules to rules based on language itself, which the model eager most.

4. Experiments

4.1. Data

We verified the effectiveness of SGSA on several standard small-scale datasets so as to simulate low-resource cases, where the Transformer was proved to lead to poor syntax. To compare with previous syntax-dependent work, following [21], we take News Commentary v11 (NC11) English→ German (En-De) and German→ English (De-En) datasets, validation and test sets are newstest2015 and newstest2016 respectively. For a standard lower source scenario, we train our model on WMT18 English–Turkish (En-Tr) set, evaluate on newstest2016, and test on newstest2017. We also carried out experiments on two other widely-used IWSLT14 De-En and IWSLT15 English→ Vietnamese (En-Vi) translation datasets, to further measure the applicability of the approach in the standard translation tasks. For the IWSLT14 De-En dataset, we follow the setup of [22], randomly sample 7K sentences from the training data to form the verification set, and test on the concatenation of multiple sets which is of similar size to the validation set. For the IWSLT15 En-Vi dataset, we use tst2012 and tst2013 as development data and test data, respectively. Table 1 shows the statistics of the datasets.

Table 1
Statistic of the datasets.

Datasets	NC11	WMT18En-Tr	IWSLT14De-En	IWSLT15En-Vi
Train	226822	207373	160239	133314
Valid	2169	3000	7283	1553
Test	2999	3007	6750	1268

4.2. Model settings

Our implementation is based on Fairseq [23]. We use `transformer_iwslt_de_en` architecture, which includes a six-layer model with 4 attention heads, embedding size 512, and FFN layer dimension 1024. In training step, we use Adam [24] to optimize the network with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and `weight-decay` = 0.0001. The learning rate scheduler is `inverse-sqrt`, where `warmup-init-lr` = 10^{-7}, `warmup-updates` = 4000 and `max-lr` = 0.001. We limit the maximum words number of input/output to 4096 per GPU, and batch sentence pairs according to sentence length. We accumulate every two steps to obtain a better batching. Dropout 0.3 and label smoothing 0.1 are used to increase the expressiveness of the model. The numbers of byte pair encoding (BPE) merge operation for IWSLT tasks and WMT18En-Tr are 15k and 16k respectively. For NC11 task, we use 8k BPE vocabulary.

For evaluation, to ensure the validity of the results, we average 5 checkpoints which are saved per training epoch as the inference model, and the beam size is set to 5. We use `multi-bleu.Perl`[1] to score the generated sentences case-insensitive for a fair comparison with previous work.

We report various results that rely on different dependency parsers for completeness. Three influential NLP toolkits that support parsing are described as follows:

- **spaCy** The spaCy (v3.2)[2] adopts ClearNLP Dependencies[3] (including 45 dependent tags), supports dependency parsing in 18 languages, with parsing accuracy of 90% for both English and German.
- **Stanza** The Stanza (v1.1) [25] is trained on the Universal Dependencies treebanks and other multilingual corpora, adopts Universal Stanford Dependencies [26] (including 61 dependent tags) and supports 66 human languages. Its parsing accuracy of English, German, Turkish and Vietnamese are 83.57%, 80.61%, 64.5% and 48.16% respectively.
- **HanLP** The HanLP (v2.1) [27] is another toolkit that adopts Universal Stanford Dependencies and supports 104 languages. It classifies languages other than Chinese into the multilingual category, including English, German, Turkish and Vietnamese, with an average parsing accuracy of 79.1%.

4.3. Main results

In this section, we evaluate the proposed approach on several standard translation tasks, as listed in Table 2.

It can be seen that Transformer has obvious advantages over the previously CNN and RNN-based models [21,34], which provides us with a strong baseline. Clearly, traditional methods do not achieve remarkable results when applied to Transformer. The MULTI-TASK [34], which shares a model to parse and translate source sentences, only attains comparable performance with the baseline. Following [40], introducing linguistic features such as syntactic lemmas and dependency labels into Transformer embedding does not bring prominent advantages. On the other hand, LISA [8] and PASCAL [14], which combine syntactic knowledge with self-attention lead to obvious improvement, validating the effectiveness of syntactic localness modeling in self-attention networks. Our SGSA outperforms all other methods, gains from +1.12 to +3.88 BLEU points compare with the Transformer baseline, achieves the highest score among all source-side syntax-aware methods.

Table 2
BLEU scores on the datasets. Bold indicates statistical significance ($p < 0.01$) against the Transformer baseline via bootstrap re-sampling [28]. Δ represents the improvement over the Transformer baseline. Normal-before (NB) means that different attention representations are normalized first and then fused. (‡) from [14].

Models	NC11		WMT18		IWSLT14	IWSLT15
	En-De	De-En	En-Tr		De-En	En-Vi
				Lightweight Conv [29]	34.80	
				Multi-agent [30]	35.56	
				Tied-Trans [31]	35.52	
				Loss to teach [32]	34.80	
Bastings [21]	16.1			Role-interactive [33],	34.74	
Mixed Enc.[34]			9.6	Variational-att [35]	33.68	
Multi-Task [34]			10.6	Bert-fused [36]	**36.11**	
Transformer	25.0	26.6	13.1		34.09	30.76
+LISA‡	25.3	27.1	13.6	ELMo [37]		29.30
+MUL-TASK‡	24.8	26.7	14.0	CVT [37]		26.90
+S&H‡	25.5	26.8	13.0	Seq2Seq-OT [38]		26.90
+PASCAL‡	25.9	27.4	14.0	C-MLM [39]	35.63	31.51
SGSA(NB)	25.36	28.31	16.86		34.99	31.11
SGSA	**26.12**	**28.46**	**16.98**		35.79	**31.89**
Δ	+1.12	+1.86	+3.88		+1.70	+1.13

More broadly, SGSA also shows strong competition compared with other well-designed NMT models. Among them, the latest BERT-fused [36] and C-MLM [39] achieve state-of-the-art results on IWSLT14De-En and IWSLT15En-Vi datasets respectively. We also display the results in Table 2 as a reference. It should be noted that both BERT-fused and C-MLM rely on the pre-training model BERT, with large parameters and a long training time, while SGSA does not introduce additional parameters. On the other hand, the BERT-fused method brings a 40% to 49% inference time increase compared with the Transformer baseline, its counterpart SGSA has no extra inference time cost. The relevant statistical results are summarized in Table 9. We also try to normalize the attention representations first and then fuse them (normal-before, NB for short), find that though this method also promotes the translation quality, its improvements are not as obvious as that of SGSA. More investigation on this point is left as future work.

In short, with no additional parameters and time overhead, our model robustly exploits syntactic information while producing high-quality translations, has achieved substantial gains in translation tasks with strict syntax norms. It is also quite competitive with the latest state-of-the-art models which incorporate pre-training features.

4.4. Ablation study

In this series of experiments, we evaluate the impact of different model components. First, we explore the impact of different parsers on translation results. Then, we examine whether it is necessary to apply syntax-aware as a bias that highlights syntax to all the layers, while checking which layers of self-attention networks benefit most from the DMSRs. Moreover, we investigate the effect of using different strategies to fusion the DMSRs. Finally, we compare other model details, such as the super parameters and different word granularity.

[1] https://github.com/mosessmt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[2] https://spacy.io/

[3] https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md

Table 3
The results of utilizing different parsers on IWSLT14 De-En and IWSLT15 En-Vi datasets. "Trans" stands for the Transformer baseline.

Parsers	IWSLT14De-En		IWSLT15En-Vi	
	Valid	Test	Valid	Test
spaCy	36.73	35.46	27.85	**31.67**
Stanza	36.81	**35.53**	27.88	31.21
HanLP	36.85	35.41	26.79	31.16
Trans	35.27	34.09	27.03	30.76

Table 4
The results of applying syntax-aware masking to highlight syntax at different encoder layers on IWSLT15 En-Vi dataset. "Trans" stands for the Transformer baseline.

Layer	spaCy		Stanza		Layer	spaCy		Stanza	
	Valid	Test	Valid	Test		Valid	Test	Valid	Test
1	27.85	31.67	27.88	31.21	1–2	27.45	**31.80**	27.71	**31.72**
2	27.83	**31.82**	27.86	31.35	1–3	27.40	31.68	27.86	31.57
3	27.85	31.81	27.67	**31.72**	1–4	27.20	31.60	27.27	30.93
4	27.07	31.09	27.00	31.30	1–5	26.51	30.64	26.63	30.36
5	25.94	30.00	27.16	31.14	1–6	26.28	30.50	26.56	30.14
6	24.72	27.15	26.85	31.04	2–3	27.58	31.79	27.53	31.50
Trans	27.03	30.76	/	/	4–6	25.99	29.81	27.09	30.81

Table 5
The results of applying different parsers on IWSLT15 Vi-En and WMT18 Tr-En datasets. Layer indicates highlight syntax at different layers.

Parsers	Layer	IWSLT15Vi-En		WMT18Tr-En
		Valid	Test	
Trans		27.11	30.94	19.27
spaCy	1	27.19	31.07	19.04
	2	26.81	30.65	18.91
	3	27.23	30.95	19.04
Stanza	1	27.48	30.68	19.26
	2	27.01	30.71	18.97
	3	27.36	30.79	19.15

Performance by different parsers. Since the model utilizes the source sentence syntactic information as a guiding basis to enhance machine translation, intuitively, the performance of the parser plays an important role in terms of translation quality. Thus we first apply parsers with different accuracy for verification experiments. The results are shown in Table 3, where we highlight the syntax only in the first encoder layer.

As can be seen, with the support of syntactic information, the model obtains consistent improvements over the Transformer baseline, which proves the effectiveness of the syntax-aware self-attention. Among the three parsers, spaCy and Stanza with higher parsing accuracy achieve the best BLEU scores on En-Vi and De-En tasks respectively, while HanLP gets relatively modest improvement compared with the first two parsers. This is in line with common sense, as relying on the source sentence syntax, high-quality parsing results will bring high benefits. The results also indicate the importance of the parser to the final result, which we will make deep analysis in subsequent sections.

Layers to highlight syntax. Recent works show that different Transformer layers tend to capture different features [41], thus it is essential to explore the applicable scope of syntactic information. We choose spaCy and Stanza with better performance as parsers and further apply syntax-aware masking to different combinations of layers, as shown in Table 4. Obviously, incorporating syntactic information into the first three layers is more beneficial. In contrast, incorporating syntactic information into the higher layers does not take significant improvement (4,5,6), and even degenerates the performance (spaCy 5,6). In brief, the performance generally goes down with the increase of layer height when integrating syntactic knowledge in only one layer. This corroborates the conclusion of [41] that the shallow Transformer layers tend to focus on the word to be translated itself rather than its context, while high layers are inverse. Explicitly enforcing syntactic dependency in the shallow layers is beneficial to better word representation, and further improve the translation accuracy, yet also, inversely, implementing the same operation at high layers that tend to encode more contextualization is more likely to disrupt the model representations.

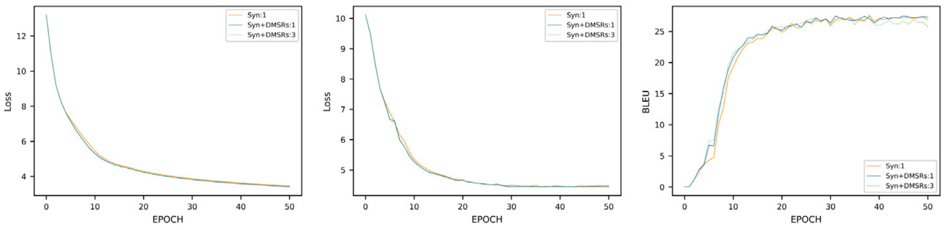
The same trend of multiple syntax-aware layers confirms this conclusion: stacking shallow layers also obtains competitive performance (1–2, 1–3, 2–3), vice versa, the downtrend is also manifested when increase layers from bottom to top (1–4, 1–5, 1–6, 4–6). In addition, multiple syntax-aware layers have not appeared to be more advantageous than single. We speculate that stacking multiple layers will cause duplication of information and easy to introduce noise, which prevents the further leap of model performance. Besides, it is worth noting that though spaCy is more accurate and generally beats Stanza yields higher BLEU scores, using Stanza as a parser seems more robust, its fluctuation range of results (31.72→ 30.14) is not as large as spaCy (31.82→ 27.15). Accordingly, we also perform further ablation studies in the following experiments.

Furthermore, combined the overall results with the finding of [14] who adjusts the self-attention weight in the first encoder layer according to the distance of each token from the position of its dependency parent, we deduce that the proposition *shallow layers is more suitable for syntax integration* is only relative to the high layers, while Transform lacks syntactic knowledge as a whole.

Performance of low resource source language. The results of the aforementioned experiments have reflected the differences in the parsers. In order to further reveal this difference and explore which degree of parsing precision can actively promote the NMT models, we consider an extreme case. We reverse the translation direction of IWSLT15 En-Vi and WMT18 En-Tr tasks, take Vietnamese and Turkish without ideally-performing parsers available as the source language. The results are shown in Table 5, as spaCy has no parsing module specifically trained for Vietnamese and Turkish, we use the English pipeline (en_core_web_sm[4]) instead.

By highlighting syntax in the first three layers with good performance, firstly, spaCy provide a slight improvement in Vi-En translation in the best case, but degenerate the performance of Tr-En translation over the baseline. Secondly, Stanza failed to improve the performance of the two tasks. With the accuracy of 48.16% and 64.5% for Vietnamese and Turkish parsing, it causes degeneration in Vi-En translation, and achieves a BLEU score equivalent to the baseline in Tr-En translation. Based on the results, we hypothesize that (1) the accuracy of the parser is crucial to the translation performance, approximately 70% is the dividing line between positive and negative; (2) when the source language has no ideally-performing dependency parsers available, the parser of its similar high resource language can be taken as replacing, which can also provide considerable effectiveness; (3) although the model is sensitive to the dependency results, it has certain fault tolerance and the performance has not dropped sharply under the poor parsing results. We will continue to explore the improvement methods where the source language is without ideally-performing parsers available in future work, such as joint training of parsing and translation models.

[4] https://spacy.io/models/en/



(a) Training loss. (b) Validation loss. (c) Validation BLEU.

Fig. 6. Training/validation curves with different applying layers. Numbers represent layers, "Syn" and "Syn+DMSRs" represent highlight syntax only and highlight syntax then overlay DMSRs respectively.

Table 6
The results of applying DMSRs (the number is 2 in this series of experiments) at different layers on the IWSLT15 En-Vi dataset, with a highway network fusion strategy, while $^{\Uparrow}$ indicates that there is a significant improvement over highlighting syntax only.

Layer	spaCy		Stanza	
	Valid	Test	Valid	Test
1	27.76	**31.89**$^{\Uparrow}$	27.73	**31.61**$^{\Uparrow}$
2	27.66	31.04	27.70	**31.45**$^{\Uparrow}$
3	27.02	31.38	27.51	31.59
1–2	27.63	31.39	27.28	31.34
2–3	27.44	30.22	26.87	31.19

Layers to apply DMSRs. Recall that the positive results indicated by syntax-aware self-attention show the effectiveness of syntactic information in helping translation, however, the experimental results also show the importance of high-performance dependency parsers. Inspired by the two phenomena mentioned before, that is, the robustness of the model when using Stanza as the parser and the fault tolerance in the case of applying low resource source language. As we rely on noisy annotations from an external parser, to prevent our model from over-fitting to noisy dependencies and further improve its characterization ability, we introduce a regularization technique for the SGSA sub-layer: node random dropping described in Section 3.3, which finally produce dynamic multiple syntax-aware self-attention representations (DMSRs).

In this series of experiments, we investigate which layers would benefit more from the DMSRs. We introduce multiple attention representations by node random dropping at different combinations of encoder layers, which are selected according to previous experimental results. As shown in Table 6, clearly, compared with highlight syntax only, applying DMSRs to the first layer brings vastly improvement. The spaCy based model boosts the BLEU scores from 31.67 to 31.89, while the counterpart Stanza strengthens the translation from 31.21 to 31.61. The superior performance indicates the effectiveness of the node random dropping technique in helping translation, and echoes the fault tolerance i.e. robustness of the model we mentioned previously. Interestingly, introducing DMSRs to the second and third layers fails to facilitate translation, except for the Stanza-based special case in the second layer. We speculate that injecting noise in lower layers is easier to converge after more iterations, but not vice versa. To some extent, this is in line with the view of [42] that higher-level layers are more representative than lower-level layers — the more the iterations, the more stable the convergence.

Moreover, simply stacking and merging in multiple layers will not bring about performance improvement, but may lead to degradation. It is worse than operating in a single layer, we attribute this to error accumulation. The results of multiple layers stacking also verify the speculation that more iterations make easier convergence, where the performance of 1–2 layers generally outperforms those of 2–3 layers.

Table 7
The results that using different types of fusion strategies to integrate DMSRs (the number is 2 in this series of experiments) on IWSLT datasets.

Models	IWSLT14De-En		IWSLT15En-Vi	
	Valid	Test	Valid	Test
Transformer(base)	35.27	34.09	27.03	30.76
Average polling	36.78	35.46	27.53	31.51
Linear network	36.89	35.69	27.54	31.77
Highway network	36.91	35.79	27.76	31.89

To sum up: (1) random dropping mechanism injects more noise, while lower layers undertake more iterative optimization, which is suitable for introducing additional noise; (2) stacking in multiple layers has the risk of error accumulation, resulting in further amplification of deviation; (3) the DMSRs take the model stronger robustness and greater noise tolerance, and is more inclined to correct errors, so that the model has stronger generalization.

We also show the experiment details of the spaCy based model in Fig. 6. As observed, highlighting syntax and then applying DMSRs to the 1st layer leads to smaller validation loss and so better BLUE scores. This indicates that this setting can indeed improve the generalization ability of our model. Moreover, the curves of the 3th layer reveal large fluctuation, which is consistent with our speculation that it is difficult to converge. In the following experiments, we use spaCy as the dependence parser, highlighting syntax and then applying DMSRs in the first layer unless otherwise noted.

Fusion strategies. We introduce different fusion strategies to integrate DMSRs. Table 7 lists the contribution of each strategy. Clearly, our proposed approaches which are based on the smaller `transformer_iwslt_de_en` model have achieved consistent improvement over the Transformer base model. These results validate the effectiveness of dynamic multiple representations in self-attention networks.

Among them, the linear and highway networks outperform average pooling on test sets, showing stronger generalization, while the highway network improves the model performance most. We attribute this to that more modeling flexibility is effective for the model to give full play to its subjective initiative. The linear layer of the highway network and the linear network provide a certain degree of freedom for the model, while the nonlinear activation of the highway network further expands the flexibility. Conversely, simple averaging different parts may lose the uniqueness of each part, which is not expected by the model. In the main experiments, we highlight syntax and apply node random dropping to the first layer of the encoder, which employs a highway network fusion strategy.

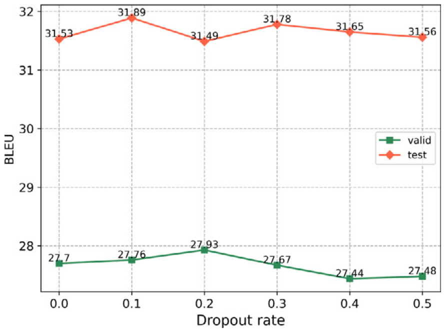


Fig. 7. Curves with different node drop rate to DMSRs.

Table 8
Results for several types of data on IWSLT datasets.

Data	Transformer(base)				SGSA			
	IWSLT14De-En		IWSLT15En-Vi		IWSLT14De-En		IWSLT15En-Vi	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
No BPE	33.16	32.12	26.67	30.32	33.29	32.44	27.23	30.58
Full	35.27	34.09	27.03	30.76	36.91	35.79	27.76	31.89
100K	32.96	31.67	26.32	30.01	34.68	33.52	26.91	30.62

Sensitivity to hyper parameters. To investigate the effect of node dropping rate, we conduct experiments on IWSLT15 En-Vi dataset with different drop probability, p ∈ {0, 0.1, 0.2, 0.3, 0.4, 0.5}. The results are shown in Fig. 7. Note that setting p = 0 refers to the case of only producing multiple syntax-aware self-attention representations based on unmodified parsing results.

As can be seen, although larger p leads to smaller validation scores, node dropping generally leads to better test BLUE scores. This shows that the drop trick can indeed make the model more robust. With the increase of p, that is, the decline of syntactic information intervention, the translation score also decreases. We fix $p = 0.1$ in the experiments unless especially specified.

Sensitivity to DMSR numbers. Thus the DMSRs play an important role in SGSA, in Fig. 8, we summarize the effect and parameters of introducing different numbers of syntax graphs (i.e. the numbers of DMSRs) in the proposed method, with a special case that the number is 0 without DMSRs.

First of all, when there are two syntax graphs in addition to original self-attention, the model gains the best result, with a gain of 0.22. Another observation is that introducing one syntax graph does not facilitate the training, this can be explained by the fact that the attention representations are not sufficiently converged. Considering the degradation of the performance as the DMSR numbers continue to increase, we argue that the benefit is not from the increased complexity of the network, but from the diversity of the representations. To this end, we perform two different node random droppings as the default setting.

Performance of extremely low resource language pairs. In order to verify the universality of the model for different granularity inputs and its performance under extremely low resources, we further conduct a series of experiments. The results are shown in Table 8, where "No BPE" means no BPE sub-word segmentation, "Full" means use the complete dataset, "100 K" means random select 100k parallel sentence pairs as training data from the original training set.

In the case of without using BPE sub-word segmentation, SGSA consistently exceeds the performance of the Transformer base model, which proves the universality of different granularity inputs and the effectiveness of our method. When randomly selecting 100k parallel sentence pairs from the original training set as training data to create an environment with extremely low resources, SGSA also vastly outperforms the baseline model, obtaining the greatest gain of 1.85 (31.67→ 33.52) on De-En task, which is higher than the improvement on the full data case. This superior result demonstrates that explicitly combining syntactic information with translation systems can provide more contextualized embeddings, and better assist the model in translation by learning extra prior knowledge, which is proved to be lacking in low resources cases.

Table 9
Comparisons of training parameters (#Param.; M) and inference time (Time; seconds).

Datasets	PASCAL	BERT-fused	Transformer(base)		SGSA	
	#Param.	#Param.	#Param.	Time	#Param.	Time
NC11En-De	52.46	51.59	50.71	23.9	43.62	21.5
NC11De-En	52.46	51.59	50.71	21.5	43.62	20.6
WMT18En-Tr	57.33	55.55	55.57	20.9	48.49	21.4
IWSLT14De-En	53.90	52.37	52.06	31.1	44.98	31.1
IWSLT15En-Vi	54.75	54.69	52.99	7.1	45.90	7.3

4.5. Training parameters and inference time

Note that SGSA is based on a much small model, which brings other advantages: fewer training parameters and shorter inference time, in other words, wider adaptability and application scenarios. Table 9 counts the training parameters and inference time. As shown, SGSA does not bring extra inference time cost, and the training parameters are more advantageous than the Transformer base model. Compared with other previous works, SGSA also shows great advantages at computational cost, in which the syntax-dependent Pascal relies on a larger model, and BERT-fused relies on the pre-trained language model, which requires additional 110M parameters from BERT. Indeed, introducing BERT also brings additional inference time, resulting in about 40% to 49% increase over the Transformer baseline.

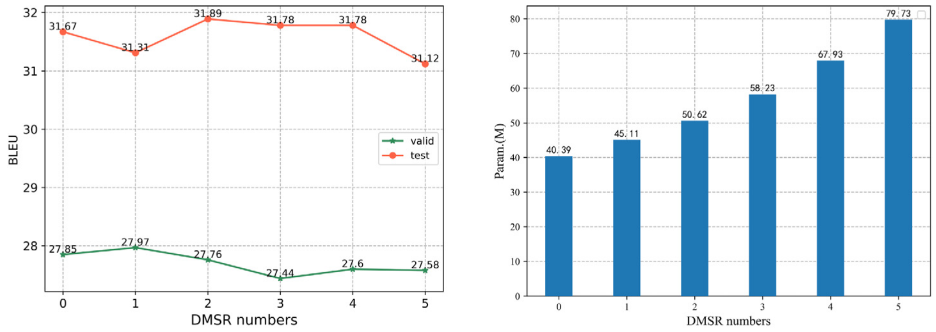
4.6. Attention visualizations

For intuitively understanding how SGSA works and affects the representation of the source sentence, we visualize the attention weight of a test sentence "Kofi is the embodiment of possibility.". Fig. 9 visualizes the weight distributions of the encoder layers under different settings.

It can be seen, the distribution of the Transformer baseline is chaotic, which is quite unsatisfactory. On the contrary, by introducing the source syntactic information as a guider, SGSA has significant improvements over the baseline: the attention distributions of the first encoder layer are arranged syntactically with diagonal as the axis, while after times iteration, the distributions of the encoder output layer are more inclined to the syntax-aware context, showing an obvious trend from self-perception to context perception, and the connections between segmented sub-words are strengthened. These results indicate that the baseline model lacks explicit guidance, while the syntax-aware localness modeling can effectively integrate syntactic knowledge into self-attention to make up for the shortcoming.

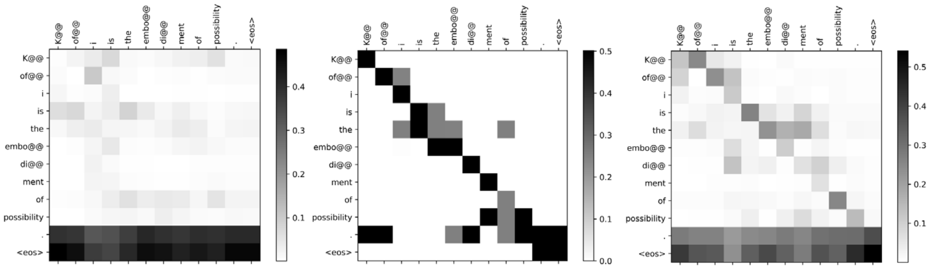
4.7. Qualitative performance

Table 10 presents examples where SGSA translates the source sentence correctly while the Transformer baseline made a syntactic error. For instance, in the first sentence, "schwierig" is used to modify the subject "Es" instead of "heraus". The sentence is actually expressing the view that the subject "it" is difficult,



(a) BLEU (b) Param.

Fig. 8. BLEU scores and model parameters of applying different numbers of DMSRs on IWSLT14 En-Vi task.



(a) Transformer base layer 6. (b) SGSA layer 1. (c) SGSA layer 6.

Fig. 9. The attention weight distributions of the encoder layers.

Table 10
Example of correct translation by SGSA. "Trans" refers to the Transformer baseline.

Src	Es stellt sich als wirklich schwierig heraus .
Tgt	It turns out to be really difficult .
Trans	It turns out it's **really hard to** get out .
Ours	It turns out to **be really hard** .
Src	ich meine , niemand hat jemals ein teilchen gesehen .
Tgt	I mean, nobody's ever seen a particle .
Trans	I mean, no one **ever saw** a particle .
Ours	I mean, nobody **has ever seen** a particle .
Src	He needed them to feel protected .
Tgt	Anh ta cần chúng để cảm thấy được bảo vệ .
Trans	**Ông** cần họ cảm thấy được bảo vệ .
Ours	**Anh ấy** cần họ để cảm thấy được bảo vệ .

not that "realize it" is difficult. In the second sentence, to express common sense and truth from the past to the present, the simple present tense should be used instead of the simple past tense. These results confirm the claim that Transformer cannot encode syntactic knowledge, while SGSA exploits self-attention to embed linguistic knowledge explicitly making great impacts on translation quality indeed, leads to consistent gains.

5. Related work

Researchers have explored various methods to improve low-resource NMT, typically including data augmentation [43,44], external knowledge integration [36,45], and domain adaptation [46–48]. In addition, the latest graph learning related works such as [49–51] are also worth learning from. Since SGSA enhances low-resource translation based on source language syntactic information, we focus on introducing related work on syntax-dependent NMT methods in this section. We discuss the related work from two aspects as follows:

5.1. RNNs with syntax

In recent years, integrating syntactic structure has been proved to be effective in improving the performance of NLP. In their course to embedding the input words, [40] add syntactic dependency labels as additional input features to improve NMT model quality. By performing parsing and translation simultaneously, [3] combine the recurrent neural network grammar into the decoder to incorporate linguistic priors. [52] introduced a top-down direction outside the bottom-up, incorporating syntax tree into the encoder bidirectionally. [53] introduce local attention, constrain attention with syntax distance, aid the model selectively focused on syntactically related words. These syntax-aware NMT models are all based on RNNs, while some recent studies have investigated the method of the Transformer model.

5.2. Transformers with syntax

[12] evaluated a much larger model with three encoders and two decoders, incorporated the dependency parse tree constructed by RNN network into Transformer. [13] started at a shallower level, concatenating the intermediate hidden representations of the dependency parser with word embedding, integrating the source syntax in a more implicit way. The drawback is that it is not suitable for sub-word segmentation, which will induce the problem of large vocabulary and unknown words. Focus on integrating target syntax into NMT, [54] interleaved words with syntax representations, accumulating gradients for

effective training. But at the same time, it is not easy to shake off the pain of a long sequence. [34] realize the integration of source syntax by means of data augmentation. One shared framework parsing and translation multitasking design is aimed at low resources, and the other mixed parsed sentence and unparsed sentence design is aimed at high resources. Recently, [55] present multi-head self-attention, a novel approach attends to phrases based on syntax and then exploits interactions among phrases to enhance the strength of structural modeling. Similarly, [14] also combined syntax with the self-attention network, which rescored the weight of self-attention by the distance of syntactic nodes. However, matrix multiplication is the main reason for the expansion of model computation.

Unlike previous work, we only strengthen syntactic perception according to the weight of attention itself, which has not been discussed yet in blending syntactic knowledge.

6. Conclusion

In this work, we present a novel method that incorporates syntactic information into the self-attention network with a syntax-aware bias, which provides a new idea for the combination of prior knowledge and NMT. Further, we utilize a dropout mechanism for dynamic multiple attention representations, and explore several strategies to integrate the multiple representations. Extensive evaluations on several standard translation tasks and the comparison with previous syntax-dependency work show that the self-attention mechanism is the best carrier for embedding syntax, for all corpus, with SGSA integrating syntax explicitly and liberally is superior to all other methods. With higher robustness and better generalization, this approach leads to consistent benefits with no training costs. Moreover, the dropping mechanism can stimulate the generalization ability of the model. For multiple representations, fusion in a deeper layer more flexible can provide more suitable information for downstream modules.

CRediT authorship contribution statement

Longchao Gong: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Data curation. **Yan Li:** Investigation, Writing – original draft, Writing – review & editing, Data curation, Visualization. **Junjun Guo:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition. **Zhengtao Yu:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Shengxiang Gao:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work was supported by National Key Research and Development Program, China (Grant No. 2020AAA0107904), National Natural Science Foundation of China (Grant Nos. 61866020, 61732005, 61972186), Applied basic Research Program of Yunnan Province, China (Grant No. 2019FB082), and Yunnan Provincial Major Science and Technology Special Plan Projects, China (Grant No. 202103AA080015)

References

[1] D. Bahdanau, K.H. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015, arXiv:1409.0473.

[2] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[3] A. Eriguchi, Y. Tsuruoka, K. Cho, Learning to parse and translate improves neural machine translation, 2017, arXiv preprint arXiv:1702.03525.

[4] X.-P. Nguyen, S. Joty, S.C. Hoi, R. Socher, Tree-structured attention with hierarchical accumulation, 2020, arXiv preprint arXiv:2002.08046.

[5] P. Koehn, R. Knowles, Six challenges for neural machine translation, 2017, arXiv preprint arXiv:1706.03872.

[6] L. He, K. Lee, M. Lewis, L. Zettlemoyer, Deep semantic role labeling: What works and what's next, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers), 2017, pp. 473–483.

[7] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, in: EMNLP, 2017.

[8] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-informed self-attention for semantic role labeling, in: EMNLP, 2018.

[9] Y.-N. Chen, W.Y. Wang, A.I. Rudnicky, Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, 2013, pp. 120–125.

[10] J. Berant, V. Srikumar, P.-C. Chen, A. Vander Linden, B. Harding, B. Huang, P. Clark, C.D. Manning, Modeling biological processes for reading comprehension, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1499–1510.

[11] H. Wang, M. Bansal, K. Gimpel, D. McAllester, Machine comprehension with syntax, frames, and semantics, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol. 2: Short Papers), 2015, pp. 700–706.

[12] S. Wu, D. Zhang, Z. Zhang, N. Yang, M. Li, M. Zhou, Dependency-to-dependency neural machine translation, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (11) (2018) 2132–2141.

[13] M. Zhang, Z. Li, G. Fu, M. Zhang, Syntax-enhanced neural machine translation with syntax-aware word representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers), 2019, pp. 1151–1161.

[14] E. Bugliarello, N. Okazaki, Enhancing machine translation with dependency-aware self-attention, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1618–1627.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, arXiv:1706.03762.

[16] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers), 2015, pp. 1715–1725.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long and Short Papers), 2018, pp. 4171–4186.

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Adv. Neural Inf. Process. Syst. 32 (2019).

[19] M.-T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, in: EMNLP, 2015.

[20] B. Yang, Z. Tu, D.F. Wong, F. Meng, L.S. Chao, T. Zhang, Modeling localness for self-attention networks, in: EMNLP, 2018.

[21] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, in: EMNLP, 2017.

[22] S. Edunov, M. Ott, M. Auli, D. Grangier, M. Ranzato, Classical structured prediction losses for sequence to sequence learning, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long Papers), 2017, pp. 355–364'.

[23] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, Fairseq: A fast, extensible toolkit for sequence modeling, in: NAACL, 2019.

[24] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2015, CoRR abs/1412.6980.

[25] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C.D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020.
[26] M.-C. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, C.D. Manning, Universal stanford dependencies: A cross-linguistic typology, in: LREC, Vol. 14, 2014, pp. 4585–4592.
[27] H. He, J.D. Choi, The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders, in: EMNLP, 2021.
[28] P. Koehn, Statistical significance tests for machine translation evaluation, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 388–395.
[29] F. Wu, A. Fan, A. Baevski, Y.N. Dauphin, M. Auli, Pay less attention with lightweight and dynamic convolutions, 2019, arXiv preprint arXiv:1901.10430.
[30] Y. Wang, Y. Xia, T. He, F. Tian, T. Qin, C. Zhai, T.-Y. Liu, Multi-agent dual learning, in: Proceedings of the International Conference on Learning Representations, ICLR 2019, 2019.
[31] Y. Xia, T. He, X. Tan, F. Tian, D. He, T. Qin, Tied transformers: Neural machine translation with shared encoder and decoder, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, no. 01, 2019, pp. 5466–5473.
[32] L. Wu, F. Tian, Y. Xia, Y. Fan, T. Qin, J. Lai, T.-Y. Liu, Learning to teach with dynamic loss functions, in: NeurIPS, 2018.
[33] D. Weissenborn, D. Kiela, J. Weston, K. Cho, Contextualized role interaction for neural machine translation, 2019, URL https://openreview.net/forum.
[34] A. Currey, K. Heafield, Incorporating source syntax into transformer-based neural machine translation, in: Proceedings of the Fourth Conference on Machine Translation (Vol. 1: Research Papers), 2019, pp. 24–33.
[35] Y. Deng, Y. Kim, J. Chiu, D. Guo, A.M. Rush, Latent alignment and variational attention, 2018, arXiv preprint arXiv:1807.03756.
[36] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, T.-Y. Liu, Incorporating bert into neural machine translation, 2020, arXiv preprint arXiv:2002.06823.
[37] K. Clark, M.-T. Luong, C.D. Manning, Q.V. Le, Semi-supervised sequence modeling with cross-view training, in: EMNLP, 2018.
[38] L. Chen, Y. Zhang, R. Zhang, C. Tao, Z. Gan, H. Zhang, B. Li, D. Shen, C. Chen, L. Carin, Improving sequence-to-sequence learning via optimal transport, 2019, arXiv preprint arXiv:1901.06283.
[39] Y.-C. Chen, Z. Gan, Y. Cheng, J. Liu, J. Liu, Distilling knowledge learned in BERT for text generation, in: ACL, 2019.
[40] R. Sennrich, B. Haddow, Linguistic input features improve neural machine translation, in: WMT, 2016.
[41] A. Raganato, J. Tiedemann, et al., 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, The Association for Computational Linguistics, 2018.
[42] A. Anastasopoulos, D. Chiang, Tied multitask learning for neural speech translation, in: NAACL, 2018.
[43] R. Sennrich, B. Haddow, A. Birch, Improving neural machine translation models with monolingual data, 2016, arXiv, arXiv:1511.06709.
[44] H. Pham, X. Wang, Y. Yang, G. Neubig, Meta back-translation, 2021, arXiv, arXiv:2102.07847.
[45] C. Baziotis, B. Haddow, A. Birch, Language model prior for low-resource neural machine translation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2020, pp. 7622–7634.
[46] J. Zeng, J. Su, H. Wen, Y. Liu, J. Xie, Y. Yin, J. Zhao, Multi-domain neural machine translation with word-level domain context discrimination, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 447–457.
[47] J. Zeng, Y. Liu, J. Su, Y. Ge, Y. Lu, Y. Yin, J. Luo, Iterative dual domain adaptation for neural machine translation, in: EMNLP, 2019.
[48] J. Su, J. Zeng, J. Xie, H. Wen, Y. Yin, Y. Liu, Exploring discriminative word-level domain contexts for multi-domain neural machine translation, IEEE Trans. Pattern Anal. Mach. Intell. 43 (5) (2019) 1530–1545.
[49] Z. Li, X. Wang, J. Li, Q. Zhang, Deep attributed network representation learning of complex coupling and interaction, Knowl.-Based Syst. 212 (2021) 106618.
[50] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, A.S. Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.
[51] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, Jkt: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523.
[52] H. Chen, S. Huang, D. Chiang, J. Chen, Improved neural machine translation with a syntax-aware encoder and decoder, in: ACL, 2017.
[53] K. Chen, R. Wang, M. Utiyama, E. Sumita, T. Zhao, Syntax-directed attention for neural machine translation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, no. 1, 2018.
[54] D. Saunders, F. Stahlberg, A. De Gispert, B. Byrne, Multi-representation ensembles and delayed SGD updates improve syntax-based NMT, in: ACL, 2018.
[55] J. Hao, X. Wang, S. Shi, J. Zhang, Z. Tu, Multi-granularity self-attention for neural machine translation, 2019, arXiv preprint arXiv:1909.02222.