Structural Supervision for Word Alignment and Machine Translation

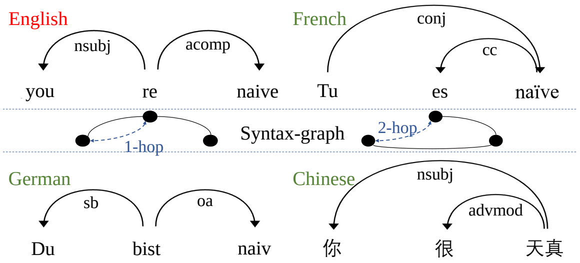
Lei Li[1,2,3], **Kai Fan**[*] [2], **Hongjia Li**[1,3], **Chun Yuan**[3]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Alibaba DAMO Academy, Alibaba Group Inc.
[3]Tsinghua Shenzhen International Graduate School, Peng Cheng Lab

Abstract

Syntactic structure has long been argued to be potentially useful for enforcing accurate word alignment and improving generalization performance of machine translation. Unfortunately, existing wisdom demonstrates its significance by considering only the syntactic structure of source tokens, neglecting the rich structural information from target tokens and the structural similarity between the source and target sentences. In this work, we propose to incorporate the syntactic structure of both source and target tokens into the encoder-decoder framework, tightly correlating the internal logic of word alignment and machine translation for multi-task learning. Particularly, we won't leverage any annotated syntactic graph of the target side during training, so we introduce Dynamic Graph Convolution Networks (DGCN) on observed target tokens to sequentially and simultaneously generate the target tokens and the corresponding syntactic graphs, and further guide the word alignment. On this basis, Hierarchical Graph Random Walks (HGRW) are performed on the syntactic graphs of both source and target sides, for incorporating structured constraints on machine translation outputs. Experiments on four publicly available language pairs verify that our method is highly effective in capturing syntactic structure in different languages, consistently outperforming baselines in alignment accuracy and demonstrating promising results in translation quality.

1 Introduction

Word alignment (Brown et al., 1993) aims to find the correspondence between tokens in a sentence pair. Neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017) works by taking an end-to-end approach to incrementally predict the target translation from a source sentence, where no explicit word alignment is required during model training or decoding. Recently, there has been an increasing interest (Zenkel et al., 2020; Chen et al., 2020, 2021; Zhang and van Genabith, 2021) in combining the two tasks through inducing accurate word alignment in neural translation models for improving translation quality.



Regardless of direction and type: (1) the dependencies between '**you**' and '**re**' and between '**re**' and '**naive**' in English match the dependencies between ‘**Du**' and ‘**bist**' and between ‘**bist**' and ‘**naiv**' in German.
(2) For English-French (Chinese) pairs, although there is no explicit dependency between '**Tu**'(**你**) and '**es**'(**很**), we can capture the implicit dependency by tracing the dependencies between '**Tu**'(**你**) and '**naïve**'(**天真**) and between '**naïve**'(**天真**) and '**es**'(**很**).

Figure 1: Similar dependencies of different languages.

Intuitively, word alignment is helpful to enforce the domain-specific terminology or improve the translations of low-frequency tokens (Song et al., 2019; Dinu et al., 2019). Also, word alignment provides supportive linguistic information on translation outputs, being useful in interactive translation with the human in the loop (Weng et al., 2019). Since the target-to-source attention in NMT models can infer rough word alignments but induce many errors with low accuracy, a number of recent works (Garg et al., 2019; Zenkel et al., 2019, 2020; Zhang and van Genabith, 2021) focus on NMT-based alignment methods which take alignments as a by-product of NMT systems.

Although NMT-based aligners have proven to be effective and achieved the State-of-the-Art alignment accuracy, they suffer from two major limitations. First, due to the autoregressive property (Sutskever et al., 2014), they (Dyer et al., 2013; Bahdanau et al., 2015; Vaswani et al., 2017; Chen et al., 2020) only leverage partial target context.

[*]Corresponding author

Findings of the Association for Computational Linguistics: ACL 2022, pages 4084 - 4094
May 22-27, 2022 ©2022 Association for Computational Linguistics

The latest works (Chen et al., 2021; Zhang and van Genabith, 2021) alleviate this deficiency to exploit both sides of the target content to compute better target-to-source attention (alignment), by abandoning autoregressive decoder and sacrificing the translation ability. In addition, there are also related works (Bastings et al., 2017; Marcheggiani et al., 2018) proposing syntax-aware NMT models without word alignment task. However, they simply utilize the syntactic structure of source tokens and ignore to capture the syntactic structure of target tokens. In summary, the syntactic structure of both source and target tokens has not been thoroughly explored to guide accurate alignments, while the similarity of dependencies across diverse languages has not been utilized for producing translation outputs with high-quality and favorable generalization capabilities. Second, they (Garg et al., 2019; Zenkel et al., 2020) typically use multi-task learning architecture to jointly learn the word alignment and translation with elaborately designed loss functions. However, this is computationally expensive for training and the internal logic between the two subtasks is not well correlated.

To alleviate mentioned problems, we propose to simultaneously consider the syntactic structure of both source and target tokens. According to the similar dependencies across language pairs, the syntactic graphs of target tokens are first sequentially inferred through introduced Dynamic Graph Convolution Networks. Hierarchical Graph Random Walks are then performed based on the built syntactic graphs at both ends, as well as the initialized multi-scale and trainable “hidden graphs” (Nikolentzos and Vazirgiannis, 2020). We found that by correlating cross-linguistic dependencies without any additional guided loss, word alignment and translation can be more effectively integrated into a unified learning framework, efficiently correlating the internal logic between subtasks while improving the interpretability of the model.

Our contributions are as follows: **(1)** We introduce Dynamic Graph Convolution Networks to sequentially infer the syntactic graphs of target tokens and further guide the word alignment learning. **(2)** Hierarchical Graph Random Walks are further performed to incorporate both local and global structural constraints for producing translation outputs. **(3)** Results on four language pairs demonstrate that our method is highly effective in such alignment- or translation-related NLP tasks, consistently outperforming baselines in alignment accuracy and translation quality.

2 Background

2.1 Word Alignment

A naive way to extract alignments from NMT models is to choose the source token with the maximum accumulated attention weight towards the current target token (Arthur et al., 2016; Hasler et al., 2018): $\gamma(t) = \underset{i\in\{1,\dots,M\}}{\arg\max} \sum_{l=1}^{N} \alpha_{t,i}^{l}$, where i is the candidate aligned source-side position. For decoding step t in layer l, $\alpha_{t,i}^{l}$ is the attention weight of the i-th position in the source, produced by an average of all the attention heads in Transformer (Vaswani et al., 2017). Although simple to implement, this method fails to obtain satisfactory alignment results (Li et al., 2019; Ding et al., 2019; Chen et al., 2020). In this work, we sufficiently exploit the similarity of dependencies between language pairs, training a novel multi-task learning framework to jointly learn translation and word alignment.

2.2 Neural Machine Translation

Let $\mathbf{x} = x_1, ..., x_M$ and $\mathbf{y} = y_1, ..., y_N$ be the source and target sentence respectively, neural machine translation models the probability of the target sentence conditioned on the source sentence: $P(\mathbf{y}|\mathbf{x};\theta) = \prod_{i=1}^{N} P(y_i|\mathbf{y}_{<i}, \mathbf{x})$, where $\mathbf{y}_{<i}$ is a partial translation from the first to (i-1)-th target tokens. Existing NMT models are generally equipped with the encoder-decoder structure. The encoder encodes the source sentence, while the decoder generates the target sentence through a target-to-source attention mechanism and performs left-to-right autoregressive decoding. In this work, we adopt Transformer (Vaswani et al., 2017) as the baseline to build our method, which is also an encoder-decoder framework while each decoder layer attends to the encoder output with multi-head attention.

3 Approach

Our work is inspired by the fact that tokens in different languages have similar dependencies under the same semantics. As shown in Figure 1, the dependencies between tokens with the same semantics in the English-German pair are highly similar, while the similarity of dependencies between English and French (Chinese) can also be implicitly captured.

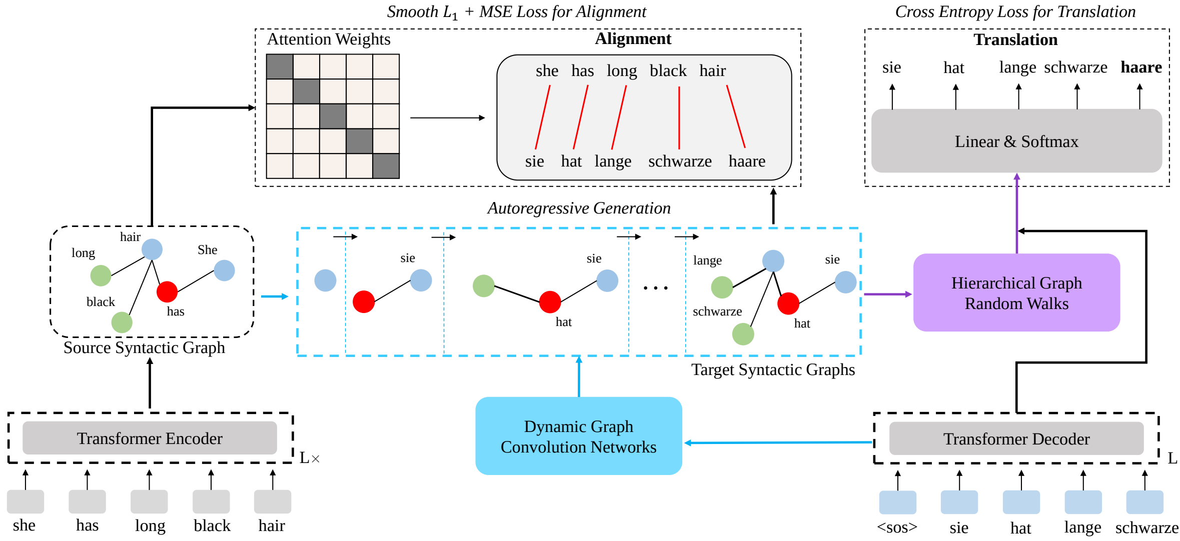

Smooth L_1 + MSE Loss for Alignment
Attention Weights
Alignment
she has long black hair
sie hat lange schwarze haare
Cross Entropy Loss for Translation
Translation
sie hat lange schwarze haare
Linear & Softmax
Autoregressive Generation
hair
long
She
black
has
Source Syntactic Graph
sie
hat
lange
schwarze
Target Syntactic Graphs
Hierarchical Graph Random Walks
Transformer Encoder
L×
Dynamic Graph Convolution Networks
Transformer Decoder
L×
she has long black hair
<sos> sie hat lange schwarze

Figure 2: The architecture of proposed multi-task learning framework. **Supervised learning tasks**: word alignment and machine translation. **Unsupervised learning task**: generation of the target syntactic graph.

We regard each token as a node, and build the edges according to the corresponding dependencies between each node to form the syntactic graphs of different languages. For instance, there is a dependency between ’you’ and ’re’, and the node ’you’ is the 1-hop neighbor of ’re’ in the built English (syntactic) graph. While there is no explicit dependency between ’Tu’ and ’es’ and we have to pass through ’naïve’ to reach ’es’ from ’Tu’, so the node ’Tu’ is treated as the 2-hop neighbor of ’es’ in the French (syntactic) graph.

3.1 Multi-task Learning

Figure 2 shows the overall architecture of proposed multi-task learning framework. We model the joint distribution of the target tokens and the target syntactic graphs by factorizing it into the product of a series conditional distributions.

$$P(\mathbf{y}, \mathbf{y}^s | \mathbf{x}^s, \mathbf{x}) = \prod_{i=1}^{N} P(y_i | \mathbf{y}^s_{\leq i}, \mathbf{x}^s, \mathbf{y}_{<i}, \mathbf{x}) \\ \times P(y^s_i | \mathbf{y}^s_{<i}, \mathbf{x}^s, \mathbf{y}_{<i}, \mathbf{x}),$$

where $\mathbf{y}_{<i}, \mathbf{y}^s_{<i}$ are partially generated target sentence and syntactic graph, and $(\mathbf{x}, \mathbf{x}^s)$ indicates the entire information from the source side. For the tokens $\mathbf{y}$, we can directly optimize translation loss. However, since we mainly focus on the word alignment dataset, we do not leverage the ground-truth of the target syntactic graph to maximize the likelihood. In order to use the supervised signal of word alignment, we propose a proxy to construct the word alignment α with graph convolution networks:

$$\alpha = \text{proxy}(\mathbf{y}^s),$$

where the proxy construction will be elaborated in the next section. Then we optimize the word alignment loss as a surrogate.

In summary, we learn three tasks simultaneously, machine translation and word alignment via supervised signals while inferring syntactic graph of the target side as a byproduct in an unsupervised way.

Specifically, our approach first build the syntactic graph of source tokens, on which basis we introduce Dynamic Graph Convolution Networks to sequentially infer the syntactic structure of observed target tokens, efficiently generating accurate alignment results which derived from the structural attention weights between both sides. To better encourage the correlation of the internal logic between word alignment and translation, Hierarchical Graph Random Walks are then performed to incorporate structural constraints for producing high-quality translation outputs.

3.2 Dynamic Graph Convolution Networks

We can first build the source syntax graph with the output representations H_e from Transformer encoder, where each node corresponds to one token representation. In particular, the adjacency matrix A_s is generated from the parsed syntactic structure, where $a_{(i,j)} = 1$ indicates there is a dependency between node i and j. Meanwhile, we initialize the rough adjacency matrix $\bar{A}_t$ containing only selfconnections for each target token. Afterwards, Dynamic Graph Convolution Networks are leveraged to adaptively adjust the graph topology for obtaining refined adjacent structures. Significantly, both masking and attention mechanisms are introduced to distinguish and re-weight observed target nodes through the captured multi-hops neighbor.

For each decoding step, masking mechanism is first built for the observed set of target nodes. For each observed token (or node), we predict a soft mask M to indicate its dependency with other observed tokens. It treats any of the observed tokens as the central node alternately, to reward its significant dependencies from multi-hops neighbor and penalize leftovers. A light-weight two-layer pooling network is used to learn the mask which could be formulated as:

$$M = f_M(A_s, \bar{H}_d, H_e),$$

where $\bar{H}_d \in R^{\bar{N} \times D}$ denotes the D-dimension features of $\bar{N}$ observed target nodes generated from Transformer decoder. The detailed network architecture of f_M can refer to the Appendix. The obtained $M \in R^{\bar{N} \times \bar{N}}$ serves as an information gatekeeper, retaining the nodes that are optimal for local aggregation with a global perspective, capturing linguistic dependencies discriminatively without compromising the topology of the syntactic graph. We will then process a graph-based information aggregation (Kipf and Welling, 2016) and proceed with a linear transformation, i.e.,

$$\bar{H}_m = W_m \cdot \left[\left(\bar{A}_t + M \right) \cdot \bar{H}_d \right] + b_m,$$

where $\cdot$ denotes the matrix multiplication and the formula in square bracket means information aggregation. In this way, the set of observed nodes and their edge connections at target side change dynamically in successive decoding steps.

In addition, an attention mechanism is introduced to re-weight and balance the captured multi-hops neighbor of each observed token. In particular, we aggregate context information by attending over the multi-hops neighbor of each node, while its updated representation is calculated by the weighted average of the connected nodes:

$$\bar{H}_a^i = \text{ReLU} \left(\sum_{y_j \in \mathcal{N}^+(y_i)} a_{ij}^{(h)} \cdot (W_a \bar{H}_m^j) \right),$$

where $j = 1, ..., \bar{N}$ and $\mathcal{N}^+(y_i)$ includes the node y_i and the nodes directly connected to y_i, W_a is a learnable parameter. Note that the attention coefficient a_{ij} is the normalized similarity between two nodes (Veličković et al., 2017) of $\bar{H}_a$ in previous decoding step, and h-hop $a_{ij}^{(h)}$ is the corresponding element of h-th power of matrix $[a_{ij}]$.

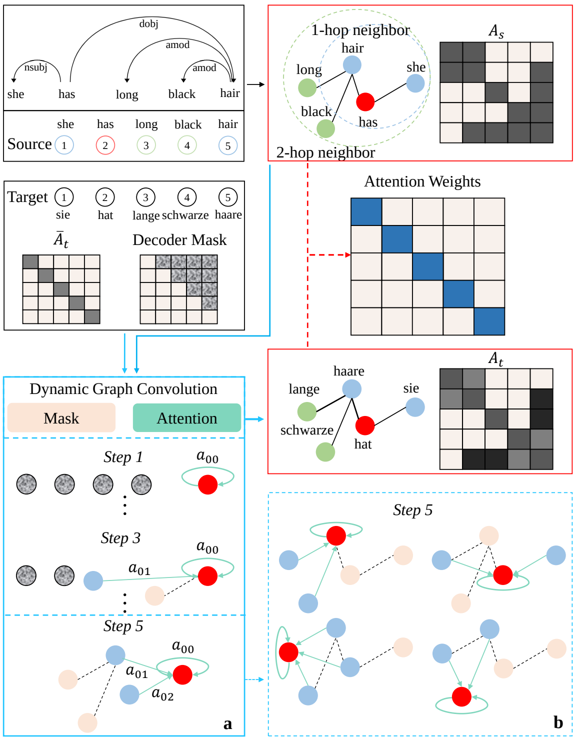


Figure 3: Structural attention based on Dynamic Graph Convolution Networks. (a) Masking and attention mechanisms during sequence decoding. (b) In each decoding step, we perform discriminative dynamic graph convolutions by treating each observed token as a central node.

Figure 3 illustrates the detailed process of introduced DGCN. The masking and attention mechanisms are iterated until the decoding process is terminated. Then we average the attention coefficients a_{ij} over all decoding steps, and normalize them to obtain the refined adjacency matrix $\tilde{A}_t$. Considering our initial intuition of the similarity for the syntactic structure at both ends, we calculate the final syntactic structure of the target sentence as follows:

$$A_t = \text{Sigmoid} \left(W_s \left(A_s + \tilde{A}_t \right) + b_s \right),$$

where W_s and b_s are learnable parameters.

Structural Attention for Word Alignment We adopt A_s and the inferred A_t to update the representation of language pairs, with the target-to-source attention in (Chen et al., 2021). The learned representation simultaneously contains the content and

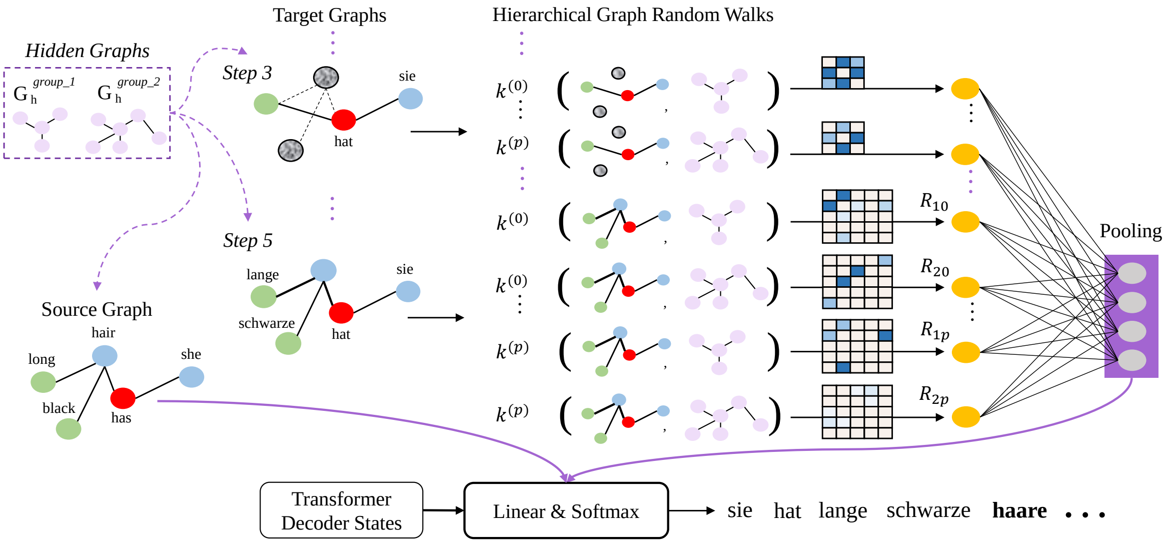


Figure 4: Overview of the introduced Hierarchical Graph Random Walks for translation.

structure information of the context for accurate word alignment. Finally, we choose the source token with the maximum attention weight towards the current target token:

$$\alpha = \text{attention}\left(A_t \cdot H_d,\ A_s \cdot H_e\right),$$
$$\gamma(t) = \underset{i \in \{1,\ldots,M\}}{\arg\max}\, \alpha_{t,i}.$$

IMPORTANT Note that even during training, we only use the ground-truth syntactic graph of source side. The syntactic graph of target side is inferred during training and its derived attention weights subsequently participate the loss calculation of word alignment task.

3.3 Hierarchical Graph Random Walks

In order to incorporate structural constraints for producing high-quality translation, we borrow the idea of (Nikolentzos and Vazirgiannis, 2020) to use a random walk kernel to capture the hierarchical representation of syntactic graphs. The random walk kernel can quantify the similarity of two graphs based on the number of common walks, adopted to effectively capture structures of the input graphs when compared against a number of trainable "hidden graphs"[1]. The adopted "hidden graphs" can learn the graph structures during training with backpropagation so that the translation outputs are highly interpretable, while the employed random walk kernel is differentiable and therefore the whole framework is end-to-end trainable. The whole process is illustrated in Figure 4.

In this work, we initialize two groups of trainable "hidden graphs" with differentiated scales, which compare the inputs using a random walk kernel to capture the structural representation of syntactic graphs both locally and globally. Consider the syntactic graph (denoted as G_d) in the decoder and a "hidden graph" G_h, their direct product $G_d^{\times}$ is a graph over pairs of nodes from G_d and G_h. We refer to the original paper (Nikolentzos and Vazirgiannis, 2020) for more details.

It has been shown that performing a random walk on the direct product $G_d^{\times}$ is equivalent to performing a simultaneous random walk on the two graphs G_d and G_h. We denote by $A_d^{\times}$ the adjacency matrix of $G_d^{\times}$, and assume a uniform distribution for the starting and stopping probabilities over the nodes of G_d and G_h. In this way, the random walk kernel will count all pairs of matching walks on G_d and G_h through the adjacency matrix $A_d^{\times}$. We then perform the P-step ($P \in \mathbb{N}$) random walk kernel which calculates the number of common walks of length p between two graphs:

$$k^{(p)}\left(G_d, G_h\right) = \sum_{i=1}^{\left|V_d^{\times}\right|} \sum_{j=1}^{\left|V_d^{\times}\right|} \left[A_d^{\times(p)}\right]_{ij}.$$

For each $p \in \{0, 1, \ldots, P\}$, a different kernel value is calculated which can be thought of as the structural representation of graph G_d. Therefore, given the two sets $\mathcal{P} = \{0, 1, \ldots, P\}$ and $\mathcal{G}_h = \left\{G_h^1, G_h^2, \ldots, G_h^K\right\}$ where G_h^1, G_h^2, ..., G_h^K denote the K "hidden graphs", we can compute one feature for each element of the Cartesian product $\mathcal{P} \times \mathcal{G}_h$, and further build a matrix $R \in \mathbb{R}^{K \times (P+1)}$

[1] Similar to the trainable "kernel" in convolution.

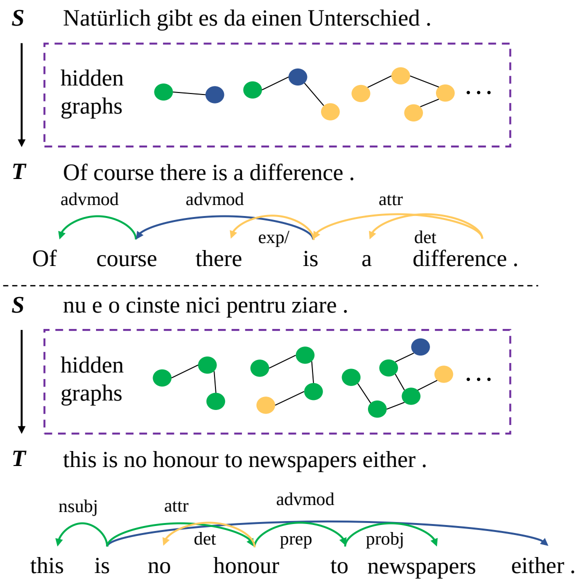


Figure 5: Visualizations of learned "hidden graphs". S: Source sentence, T: Translation output. (The ground-truth syntactic structure parsed by tools are given.)

for G_d where $R_{ij} = k^j\left(G_d, G_h^i\right)$. Finally, the matrix R is flattened as supplementary representation to incorporate structural constraints into the decoder outputs from Transformer for guiding translation outputs.

In order to capture both local and global structural information, we assign two differentiated scales (with node sizes 4 and 6) of "hidden graphs" to compare against the syntactic graphs at both ends. In the meantime, the syntactic information from both encoder and decoder are considered to access the robust and high-quality translation system. We also provide case studies of the experiments in Figure 5, demonstrating the learned "hidden graphs" can capture both the local and global dependencies of target sentences, leading to more discriminative features which are further adopted to guide the translation outputs.

4 Experiments

4.1 Datasets

We conducted our experiments on four publicly available datasets. For German-English (de-en)[2], Romanian-English (ro-en) and French-English (fr-en)[3], we followed the experimental setup in (Zenkel et al., 2020) and used the preprocessing scripts from (Zenkel et al., 2019). We also followed (Ding et al., 2019) to set the last 1K sentences of the training data before preprocessing as validation set. The Chinese-English training set is from the NIST corpora while the test set is from the v1-testset released by TsinghuaAligner (Liu and Sun, 2015). We learned a joint source and target Byte Pair Encoding (BPE) (Sennrich et al., 2016) with 10K merge operations.

4.2 Settings

We adopted parsing tools[4] to construct syntactic graphs for the language of the encoder. Both the encoder and the decoder of Transformer have 4 layers of attentions with 4 attention heads each. The embedding size and hidden states are set to 512, while the feed-forward layer has 2,048 cells. The training token-level batch size is 36K. All models were trained in both translation directions and symmetrized with *grow-diag* (Koehn et al., 2005) using the script from (Zenkel et al., 2019)[5]. We aggregated the 1- and 2-hop neighbor of each target token in proposed dynamic graph convolutions for alignment, and performed $\mathcal{P} = \{0, 1\}$-steps random walk with beam size to 4 in the decoding process of translation. Alignment error rate (AER) (Och and Ney, 2000) and BLEU (Papineni et al., 2002) are used for measuring word alignment accuracy and translation quality, respectively.

4.3 Baselines

We compare our method with two statistical baselines FAST-ALIGN (Dyer et al., 2013) and GIZA++ (Brown et al., 1993). Besides, our proposal (structure-based) is compared to several neural baselines (content-based), and all the baselines induce alignments from attention weights of content-based representation: NAIVE-ATT (Garg et al., 2019), NAIVE-ATT-LA (Garg et al., 2019), NAIVE-ATT-LA (Garg et al., 2019), SD-SMOOTHGRAD (Ding et al., 2019), ADDSGD (Zenkel et al., 2019), SHIFT-ATT (Chen et al., 2020), SHIFT-AET (Chen et al., 2020), BTBA (Zhang and van Genabith, 2021) and MASK-ALIGN (Chen et al., 2021).

NAIVE-ATT (Garg et al., 2019) induces alignments from cross-attention weights of the best penultimate layer in a vanilla Transformer.

NAIVE-ATT-LA (Garg et al., 2019) without layer selection induces alignments from attention weights averaged across all layers.

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

[3] http://web.eecs.umich.edu/~mihalcea/wpt/index.html

[4] https://github.com/explosion/spacy-models/releases

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

Method	Full	de-en				fr-en				ro-en			
		de→en	en→de	*avg.*	*bidir.*	fr→en	en→fr	*avg.*	*bidir.*	ro→en	en→ro	*avg.*	*bidir.*
Statistical Methods													
FAST-ALIGN (Dyer et al., 2013)	Yes	28.5	30.4	29.5	25.7	16.3	17.1	16.7	12.1	33.6	36.8	35.2	31.8
GIZA++ (Brown et al., 1993)	Yes	18.8	19.6	19.2	17.8	7.1	7.2	7.2	6.1	27.4	28.7	28.1	26.0
Neural Methods (**Content-based**)													
NAIVE-ATT (Garg et al., 2019)	No	33.3	36.5	34.9	28.1	27.5	23.6	25.6	16.0	33.6	35.1	34.4	30.9
NAIVE-ATT-LA (Garg et al., 2019)	No	40.9	50.8	45.9	39.8	32.4	29.8	31.1	21.2	37.5	35.5	36.5	32.7
SD-SMOOTHGRAD (Ding et al., 2019)	No	36.4	43.0	39.7	29.0	25.9	29.7	27.8	15.3	41.2	41.4	41.3	32.7
ADDSGD (Zenkel et al., 2019)	No	26.6	30.4	28.5	21.2	20.5	23.8	22.2	10.0	32.3	34.8	33.6	27.6
SHIFT-ATT (Chen et al., 2020)	No	20.9	25.7	23.3	17.9	17.1	16.1	16.6	6.6	27.4	26.0	26.7	23.9
SHIFT-AET (Chen et al., 2020)	No	**15.8**	19.2	17.5	15.4	9.9	10.5	10.2	4.7	22.7	**23.6**	23.2	21.2
BTBA (Zhang and van Genabith, 2021)	Yes	30.3	32.3	31.3	17.8	14.9	20.2	17.6	9.5	33.0	38.6	35.8	22.9
MASK-ALIGN (Chen et al., 2021)	Yes	-	-	-	14.4	-	-	-	4.4	-	-	-	19.5
Our Neural Method (**Structure-based**)													
Ours	No	16.3	**18.1**	**17.2**	**13.7**	**9.2**	**9.7**	**9.5**	**4.1**	**21.9**	23.8	**22.9**	**18.8**

Table 1: AER on the test set. The column Full denotes whether full target sentence is used to extract alignments at test time. *avg.* are the averaged AER scores of both language directions for each language pair, and *bidir.* are symmetrized alignment results. The lower AER, the better. We mark best results among all with boldface.

SD-SMOOTHGRAD (Ding et al., 2019) induces alignments from token saliency.

ADDSGD (Zenkel et al., 2019) explicitly adds an extra attention layer on top of Transformer to predict the to-be-aligned target token.

SHIFT-ATT (Chen et al., 2020) induces alignments when the to-be-aligned target token is the decoder input instead of the output.

SHIFT-AET (Chen et al., 2020) extracts alignments from an additional module with supervision from symmetrized SHIFT-ATT alignments.

BTBA (Zhang and van Genabith, 2021) predicts the current target token by paying attention to the source context and both left-side and right-side target context to produce target-to-source alignment.

MASK-ALIGN (Chen et al., 2021) extracts alignments from introduced *leaky attention* and trains with the masked language model fashion.

4.4 Alignment Results

Comparison with Baselines Table 1 compares the alignment results of our method with all the baselines. Our approach significantly outperforms both statistical and neural baselines. Specifically, it improves over GIZA++ by 2.0-7.2 AER points across different language pairs, demonstrating that building a neural aligner is better than statistical aligners. When compared with neural baselines either using guided training or without guidance, we find our proposal still achieves substantial improvements over all methods. For instance, it improves over SHIFT-AET and MASK-ALIGN by 2.4 and 0.7 individually AER points on the Romanian-English pair, indicating that the incorporation of syntactic structure achieves superior alignment results compared to these that rely only on the content of inputs.

Method	zh→en	en→zh	*bidir.*
SHIFT-ATT	28.1	27.3	20.2
SHIFT-AET	20.1	22.0	17.2
MASK-ALIGN	-	-	13.8
Ours	**18.9**	**21.2**	**13.5**

Table 2: AER on the test set of zh-en.

Besides, we also evaluate our proposal on Chinese-English pair and compare other methods in Table 2. The experimental results are highly consistent with the observations on other language pairs, demonstrating the effectiveness of alignment based on modeling dependencies and capturing structural similarities for distant language pairs.

Ablation Study Table 3 shows the ablation results on two language pairs. Our approach achieves a gain of 23.8 and 14.6 AER points with fewer parameters compared to vanilla Transformer. When considering the introduced Dynamic Graph Convolution Networks, the aggregated 1-hop neighbor can only capture the local structure, and thus the alignment accuracy is limited. In contrast, aggregating all the 1-, 2-, and 3-hop neighbor for each target node, while better capturing the global dependency, brings with it an increase of parameters and the possible introduction of noisy nodes. We finally achieve the trade-off between performance and parameter size by aggregating both the 1- and 2-hop neighbor. Notably, the accuracy of alignment slightly decreases when we remove the translation task, showing the effectiveness of our multi-task learning framework.

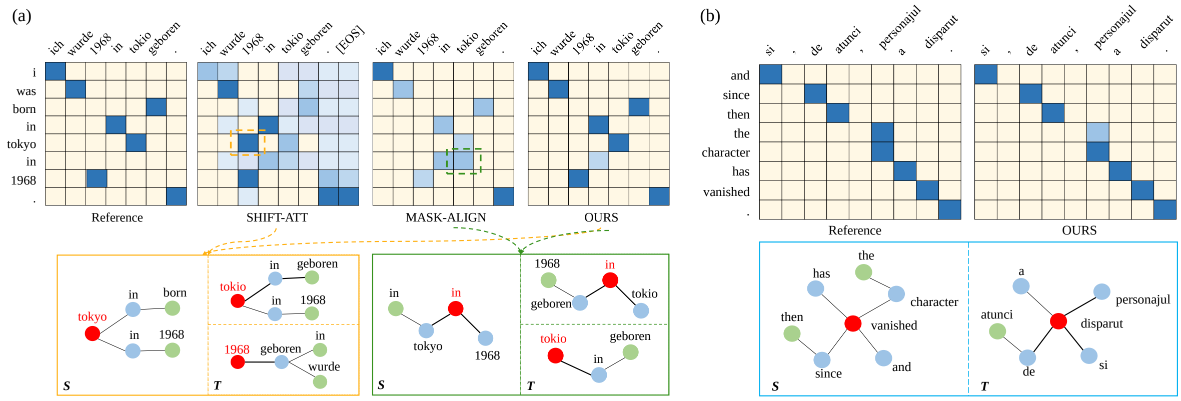


Figure 6: (a) Attention weights from different models, and visualizations of the local connection structure for important target tokens inferred by our method. Gold alignment is shown in Reference. S: Source tokens, T: Target tokens. (b) Attention weights for a symmetrized alignment example from ro-en test set. Besides, visualizations of syntactic graphs which built from the source sentence and inferred from the target sentence are given.

Method	fr-en	ro-en	# param.
Vanilla Transformer	27.9	33.4	36.8M
Ours$_{(1-hop)}$	7.6	25.8	31.9M
Ours$_{(1,2,3-hop)}$	4.4	19.3	33.2M
Ours (w/o translation)	4.3	19.5	28.5M
Ours$_{(1,2-hop)}$	**4.1**	**18.8**	32.7M

Table 3: We report the symmetrized AER on the test set. We treat vanilla Transformer (Vaswani et al., 2017) as the baseline, and Ours$_{(1,2,3-hop)}$ indicate that the introduced graph convolutions aggregate representation from all the 1-, 2-, and 3-hop neighbor. Ours (w/o translation) denotes that we remove the translation branch and only perform the training and test of word alignment.

Case Study Figure 6(a) shows the attention weights from three different models for a symmetrized alignment example from de-en test set. In this example, SHIFT-ATT puts high weights wrongly on “1968” when predicting the target token “tokyo”, while MASK-ALIGN fails to resolve ambiguity when predicting the target token “in”. In contrast, our approach produces the attention weights based on structural matching of source and target tokens, which are highly consistent with the gold alignment. Furthermore, we visualize the complete syntactic structure inferred by introduced DGCN in Figure 6(b), which could explicitly reflect the dependencies between each target token.

4.5 Translation Results

Comparison with Baselines Table 4 shows the comparison of translation quality and the corresponding decoding speed. Although this work has improved the performance of word alignment, our experiments show that the benefits from the representation of syntactic structure also extend to the translation task. Compared with (Marcheggiani et al., 2018) that only utilize syntactic structure at the encoder side, we substantially improve the performance by incorporating syntactic structure at the decoder side.

Ablation Study To investigate the effectiveness of introduced Hierarchical Graph Random Walks, we further conducted ablation experiments from two perspectives: the number of steps for random walk and the beam size for decoding. Table 4 shows the comparison results. It can be inferred that increasing the step length (e.g., $p = 2$) can improve the capability of “hidden graphs” to better capture the global structure. However, continuing to increase the step (e.g., $p = 3$) length will not always improve the performance, since it not only introduces more parameters, but also is likely to confuse the model by the complicated closed-loop structure which is prevalent in the graph network. Moreover, increasing the beam size does not bring sustainable gains, but it inevitably decreases the speed of decoding. Notably, the quality of translation significantly decreases when we remove the alignment branch, suggesting that the internal logic of both tasks are tightly correlated by exploiting the dependencies between language pairs for multi-task learning.

5 Related Works

Our work is closely related to unsupervised neural word alignment. While early unsupervised neural aligners failed to outperform their statistical counterparts such as FAST-ALIGN (Dyer et al.,

Method	de→en	en→de	fr→en	en→fr	ro→en	en→ro	$avg.speed$ (tokens/sec)
vanilla Transformer (Vaswani et al., 2017)	25.1	21.7	22.9	21.4	25.3	17.9	70.1
CNN + GCN (Marcheggiani et al., 2018)	23.4	20.3	20.7	20.1	23.8	16.4	86.5
BiRNN + GCN (Marcheggiani et al., 2018)	23.9	20.6	21.2	20.5	24.3	17.0	82.3
Ours$_{(p=1,beam=4)}$	25.7	**22.7**	**24.2**	22.3	26.2	**18.9**	68.2
Ours$_{(p=2,beam=4)}$	25.5	22.6	**24.2**	**22.4**	26.2	18.5	66.7
Ours$_{(p=3,beam=4)}$	**25.9**	**22.7**	23.6	22.0	25.8	18.8	66.3
Ours$_{(p=1,beam=3)}$	25.5	22.4	23.9	22.0	26.1	18.6	68.9
Ours$_{(p=1,beam=5)}$	25.7	22.5	24.0	22.2	**26.3**	18.3	66.5
Ours$_{(p=1,beam=4)}$ (w/o alignment)	25.5	22.1	23.4	21.7	25.5	18.3	76.7

Table 4: Comparison of BLEU scores and the averaged decoding speed tested on test sets of three language pairs. p refers that a p-step random walk is performed during the decoding process, while $beam$ is the beam size.

2013) and GIZA++ (Och and Ney, 2003), a lot of latest works (Li et al., 2019; Garg et al., 2019; Zenkel et al., 2019, 2020) have made significant progress by inducing unsupervised neural aligners from NMT to produce better word alignments. Significantly, BTBA (Zhang and van Genabith, 2021) and MASK-ALIGN (Chen et al., 2021) leverage the both side content information of the decoder, sacrificing the ability of translation.

Our work is also related to syntax-based or Transformer based neural machine translation models which have shown large advantages on a myriad of datasets. (Bastings et al., 2017) incorporated syntactic structure into the encoder of NMT model and proposed syntactic GCNs. (Marcheggiani et al., 2018) refined the above work to inject a semantic bias into sentence encoders. Transformer based NMT models (Vaswani et al., 2017; Hasler et al., 2018) attribute their superior performance to the multi-layer and multi-head self-attention architecture. (Garg et al., 2019) trained the Transformer to jointly learn word alignment and translation through multi-task learning based on existing token aligners such as GIZA++ (Och and Ney, 2003). Our work differs from prior studies in that we simultaneously incorporate the syntactic structure into both encoder and decoder to tightly correlate the internal logic of word alignment and machine translation for multi-task learning. To the best of our knowledge, this is the first work that incorporates syntactic structure based constraints into the decoder of NMT models.

6 Conclusion

We propose a multi-task learning framework that tightly correlates the internal logic of word alignment and machine translation, by fully exploits the syntactic structure of both source and target tokens and the similarity of dependencies at both ends. Experiments show that our proposal achieves the new State-of-the-Art results among all neural methods in word alignment, while producing high-quality translations. We leave it for future work to extend our study to more downstream tasks and systems in natural language processing.

Acknowledgements

This work was supported by NSFC project Grant No. U1833101, SZSTI Grant No. JCYJ20190809172201639, SZSTI Grant No. WDZC20200820200655001, and Alibaba DAMO Academy.

References

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chi Chen, Maosong Sun, and Yang Liu. 2021. Mask-align: Self-supervised neural word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4781–4791, Online. Association for Computational Linguistics.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Diego Marcheggiani, Jasmijn Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.

Giannis Nikolentzos and Michalis Vazirgiannis. 2020. Random walk graph neural networks. *Advances in Neural Information Processing Systems*, 33:16211–16222.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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

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

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Rongxiang Weng, Hao Zhou, Shujian Huang, Lei Li, Yifan Xia, and Jiajun Chen. 2019. Correct-and-memorize: Learning to translate from interactive revisions. *arXiv preprint arXiv:1907.03468*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Jingyi Zhang and Josef van Genabith. 2021. A bidirectional transformer based alignment model for unsupervised word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 283–292, Online. Association for Computational Linguistics.

A Detailed Network Architecture of f_M

For each observed token from the target side, we learn a soft mask M to predict its dependency with other observed tokens by a light-weight network:

$$\begin{aligned}\bar{h}_d &= M_{ean}P_{ooling}(\bar{H}_d + A_s \cdot H_e),\\ \hat{M} &= (\bar{H}_d \cdot W_d) \otimes \bar{h}_d,\\ M &= \text{Sigmoid}\left(M_{ax}P_{ooling}(\hat{M})\right),\end{aligned}$$

where $\otimes$ denotes the element-wise multiplication, and W_d is a trainable matrix.

B Translation Results

Case Study We provide the translation results among different variants of our proposal in Figure 7.

S Damit ist unsere Aussprache über den Stand der Europäischen Union geschlossen .

↓

T (1) *w/o* random walk

Our European Union state debate on the is concluded .

(2) random walk only in decoder

This concludes our debate on the the European Union state.

(3) **Ours** (random walk in encoder + decoder)

This concludes our debate on the state of the European Union.

R The debate on the state of the European Union is closed .

Figure 7: Translation outputs generated by our methods. S: Source sentence, T: Translation output, R: Ground-truth translation. (1) No graph-based random walk is performed for translation task. (2) Graph-based random walk is performed only on the inferred syntactic graphs of the decoder. (3) Graph-based random walks are performed on the syntactic graphs of both the encoder and the decoder.