Improving Neural Machine Translation by Efficiently Incorporating Syntactic Templates

Phuong Nguyen[1], Tung Le[2,3], Thanh-Le Ha[4], Thai Dang[5], Khanh Tran[5], Kim Anh Nguyen[5], and Nguyen Le Minh[1]

[1] Japan Advanced Institute of Science and Technology
[2] Faculty of Information Technology, University of Science, Ho Chi Minh, Vietnam
[3] Vietnam National University, Ho Chi Minh city, Vietnam
[4] Karlsruhe Institute of Technology
[5] Vingroup Big Data Institute, Vietnam
{phuongnm, nguyenml}@jaist.ac.jp, lttung@fit.hcmus.edu.vn, thanh-le.ha@partner.kit.edu, {v.thaidt4, v.khanhtv13, v.anhnk9}@vinbigdata.org,

Abstract. In the success of Transformer architecture in Neural Machine Translation, integrating linguistic features into the traditional systems gains a huge interest in both research and practice. With less increase in computational cost as well as improving the quality of translation, we propose an abstract template integration model to intensify the structural information in source language from syntactic tree. Besides, the previous works have not considered the effect of the template generating mechanism, while this is an essential component of template-based translation. In this work, we investigate various template generating methods and propose two prominent abstract template generation techniques based on the POS information. Together with the strength of Transformer, our proposed approach allows to effectively incorporate and extract the linguistics features to enrich the information in encoding phase. Experiments on several benchmarks prove that our approaches achieve competitive results against the competitive baselines with less effort in training time. Furthermore, our results reflect that syntactic information is the rich fertile ground to have benefited greatly in neural machine translation. Our code is available at https://github.com/phuongnm-bkhn/multisources-trans-nmt.

Keywords: Machine translation · Syntactic template · Transformer.

1 Introduction

Neural machine translation models (NMT) have been taken much attention in machine translation domain. The key idea of NMT is based on encoder-decoder models which have been upgraded with Transformer models [11]. One of the promising research directions is to incorporate linguistic information into the encoder representation and guide the decoder to enhance the generation. The recent work on using target syntactic templates with NMT [15] has shown that

utilizing soft template prediction could lead to large translation gains (Figure 1). However, the authors only considered the syntactic template of target side that is generated by a pruning technique based on length of target sentence. Obviously, the performance of these approaches is based on the quality of target template prediction from the source sentence. However, extracting the general instruction of target sentences in low-resource language is a great challenge in both research and practice.

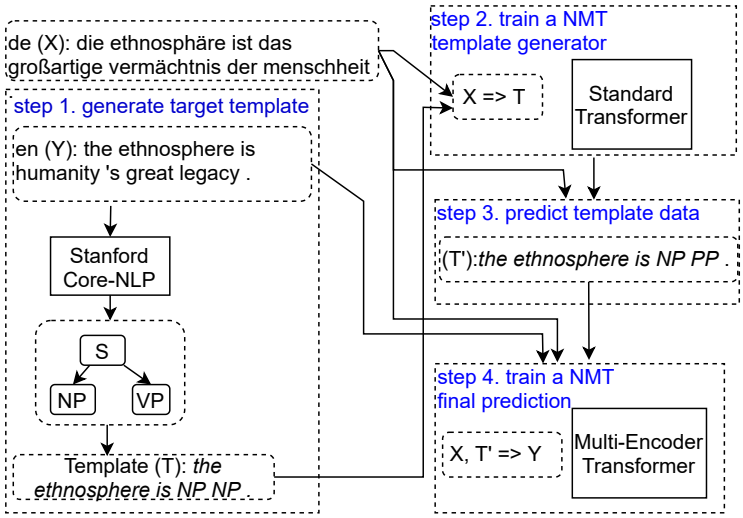

Fig. 1. Flow of NMT system using soft target template.

On the other hand, many previous works have claimed that using source tree (phrase) or tree structure in the encoder component by modifying the self-attention layers of Transformer architecture would help to improve the translation quality [2, 16, 4, 6]. Indeed, it is more straightforward to extract and integrate the structure of source sentences into the encoder phase. In this paper, inspired from the work of [15], our work is proposed to answer some natural questions: *(1) how the different kind of templates affect to the performance - template on abstract or detailed levels? (2) what is the better between two ways of integrating syntactic information - on the source side or on the target side? (3) how to inject syntactic information into the NMT model effectively?*

In the most related works against our approach, [15] showed that soft target template is potential to guide the decoding process for improving the performance of NMT systems. While they encoded the target template as a second source representation, there are, however, not any constraints related to the POS tags in the templates (e.g. NP, VP) in decoding phase. With our observation in their approach, the generated soft templates may adversely affect the translation quality. The enhancement of this model comes from the generalization of tem-

plate prediction phase. Obviously, it depends on both the performance of parser in target side and the strength of predictor. In addition, [15] need to perform an external component in order to produce the template. This becomes a 2-fold process and might suffer from error propagation.

In less modification and promising performance, we propose a direct approach to consolidate the context of source sentence via structural information in the source template instead of the target template. With our proposed model, the process of learning and integrating the syntactic structure into machine translation model is done continuously. On the other hand, we also inherit advanced technologies in the language understanding. Without any external components, our model avoids the error transmission against the previous approach in parsing and predicting the target template. In our model, the structure of source sentence comes from its natural characteristics from the syntactic parser. With our integration, our model is powerful to intensify the structure of sentence that is highly useful for translation. Obviously, it exists the corresponding structure between the source and target sentence. Therefore, our proposed approach with the intensification of source template is the promising guidance for translating phase. Especially, in the case that the target side is in low-resource languages, our proposed method is more effective and applicable than previous works utilizing the target template.

Besides the side of template, the structure of template is critical to maintain the meaning and syntactic of sentence. Therefore, to reflect the effect of template extraction into the NMT system, we conducted experiments using different kinds of templates from an abstract level containing constituent POS tags of a sentence to a detailed level containing a mixture of both POS tags and words. To prove the strength of our proposed model, this investigation is simultaneously done in both target-based and source-based approaches. Experimental results in several popular MT benchmarks showed that our approach achieves the promising results against both competitive baselines and target-based method. Especially, through our detailed comparison, it also emphasize the strength of our proposed model in low-resource language.

2 Related Works

Many works have been considered to utilize the linguistic structure representation for improving NMT, both in the encoder and decoder components. [14] indicates that a source phrase representation can be applied for boosting the performance of NMT. [13] introduces tree encoder architecture for Transformer. The works presented in [15, 10] demonstrate that the use of soft template prediction can improve NMT. Besides, Template-based machine translation also typically are applied in the Semantic parsing field to deal with the complicated logic syntax [1], the various entity names problem [5], or support to generate response in a Dialog system [3]. Based on the success of previous works in this area, our work is inspired by the approach of [15].

3 Proposed Template Integration

In this section, we would like to sketch the main ideas of how to extract syntactic templates to be used in NMT architectures. In the work of [15], the templates are generated from the syntactic tree of the target sentences based on some length-based heuristic. Based on observation and assumption from the linguistic features, we also propose two other approaches to generate templates from the syntactic tree. Besides, our proposed techniques are deployed on both the source and target sides to evaluate the effect of template extraction and the strength of proposed frameworks.

3.1 Template Generating Methods

Given the syntactic tree, each POS tag is a non-terminal node, and each leaf node is a terminal node containing one tokenized word. Based on one of the below methods, some non-terminal nodes may be pruned by removing their child nodes and become terminal nodes. Finally, the template is the list of all leaf nodes of the pruned syntactic tree.

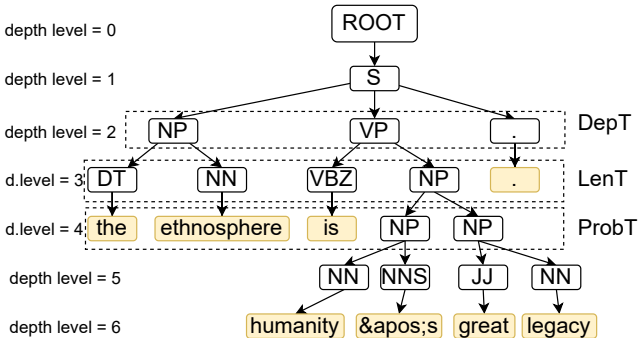

Fig. 2. Different depth levels in Syntactic Tree of three types of Template: LenT, ProbT, DepT.

*Length-based Template (*LenT*)* is proposed by [15], the template depends on length of input sentence: $d = \min(\max(L \times \lambda, \gamma_1), \gamma_2)$ where d is the depth level for pruning; γ_1, γ_2 is lower and upper bound depth that are extracted from parsed syntactic tree in each sentence, respectively; λ is hyper-parameter reflects the dependency between the pruning depth level with the length of sentence. For example in Figure 2, the sentence length (L) is 8, with $\lambda = 0.15$, $\gamma_1 = 3$ and $\gamma_2 = 6$, therefore, the pruning depth level is 3.

*Probability-based Template (*ProbT*)* is based on the average of probabilities of POS tags at each tree level to choose the best depth level for representing the template. Coming from the lack of POS consideration in previous work, we propose to utilize these information to extract the abstract template via the distribution of POS tags in languages. In particular, we analyze to obtain the probabilities of all POS tags in the training data. In our assumption, the higher probabilities the level obtains, the less noises we avoid against the rare POS tags (e.g. CVZ) in the original template.

$$d^* = \underset{\gamma_1 \leq d \leq \gamma_2}{\text{argmax}}(\text{mean}(p_{d,1}, p_{d,2} \ldots, p_{d,i}))$$

where $p_{d,i}$ is the probability that of POS tag i^{th} in the depth level d. After this step, we find the best depth (d^*) for pruning and get the soft template. For example in Figure 2, d^* is chosen in the range from 3 to 6, and the depth level 4 is the level containing the most frequent POS tags.

*Depth-based Template (*DepT*)* is extracted from the first depth level of *simple declarative clause* tag (i.e *"S"*). We aim to get the highest abstract level of template for the sentence POS tags representation. The depth level for pruning (d) is fixed by formula: $d = d_S + 1$ where d_S is the first depth level of sentence POS tag. With this method, we expect that it is easier than others to model generalize structure of a natural sentence. For example in Figure 2, $d_S = 1$, and this template is generated from the syntactic tree depth level 2.

3.2 Template Sides

Besides the template extraction techniques, we also emphasize the importance of template sides in the machine translation. Specifically, given a pair of sentences (X, Y), there are two different types of templates: templates of the source sentence (X) or template of the target sentence (Y).

Source Template. In this setting, the source template is generated from a syntactic tree based on one of the template generating methods. After that, both source sentence and the template are used to decode the target sentence.

Target Template. In this setting, similar to [15], the translation process is split into 2 phases: (1) decoding the target template from the source sentence; (2) incorporating decoded target template with the source sentence to decode the target sentence. For example, the translation of this German→English sentence pair (X, Y): *"die ethnosphäre ist das großartige vermächtnis der menschheit ."→"the ethnosphere is humanity 's great legacy ."* will be split into two steps: decoding $X \rightarrow T$ and then decoding $(X, T) \rightarrow Y$ where T is *"the ethnosphere is NP NP ."*

4 Model Architecture

Transformer (baseline) Together with the recent works, we consider Transformer model [12] as a competitive baseline for the machine translation task.

Transformer Multi Encoders (TME) For incorporating target template information, we re-implement a architecture similar to [15]). The model consists of

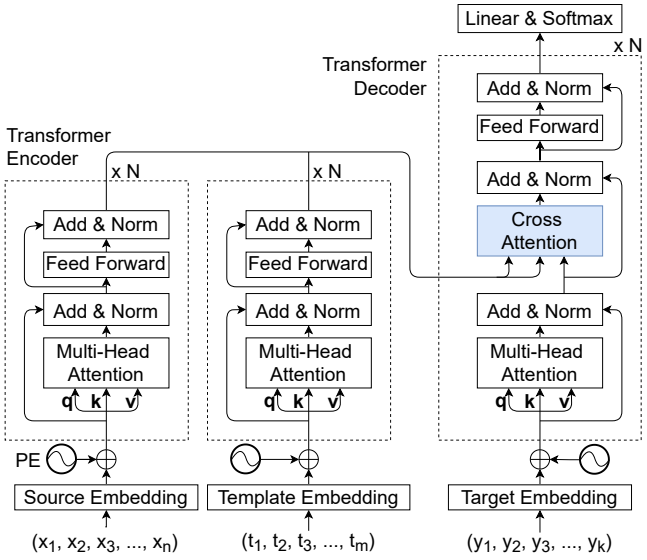

Fig. 3. Transformer model with multi Encoders.

the source encoder, template encoder, and target decoder components which are based on the Transformer architecture. The Cross–Attention learns the attention scores of both source and template, separately.

$$\boldsymbol{H}^{xy} = \text{Attention}(\boldsymbol{H}^{x}; \boldsymbol{H}^{x}; \boldsymbol{H}^{y}) \tag{1}$$

$$\boldsymbol{H}^{zy} = \text{Attention}(\boldsymbol{H}^{z}; \boldsymbol{H}^{z}; \boldsymbol{H}^{y}) \tag{2}$$

$$\boldsymbol{r} = \text{Sigmoid}(\boldsymbol{W_1}\boldsymbol{H}^{xy} + \boldsymbol{W_2}\boldsymbol{H}^{yz}) \tag{3}$$

$$\boldsymbol{H}^{y} = \boldsymbol{r} \cdot \boldsymbol{H}^{xy} + (\boldsymbol{1} - \boldsymbol{r}) \cdot \boldsymbol{H}^{zy} \tag{4}$$

where $\boldsymbol{H}^{xy}$, $\boldsymbol{H}^{zy}$ are the incorporating hidden states of the *source-target* and *template-target*, respectively; $\boldsymbol{r}$ is the impacting coefficient of source and template; $\boldsymbol{H}^{y}$ is the hidden state of the target language that contains both source and template information; Attention is the function similar to [12] that flows information from encoder to decoder.

Drop Template Mechanism. We follow the observation of [15] that the model achieves better performance when dropping the soft target template by a dropping probability (e.g. 0.5). We also randomly replaced the Equation 4 by $\boldsymbol{H}^{y} =$

$\boldsymbol{H}^{xy}$ as in the baseline model. It means ignoring template information and keeping original source sentence.

Source Template Concatenation (STC). Intuitively, the implicit relations between the source and the template may be useful for the translation process. Therefore, we use a simple method to concatenate the source and the template via a *[SEP]* token and then proceed with the concatenation as a normal input in our Transformer model (Figure 4). In this way, the relationship between source and template can be learned in the self-attention mechanism of the Transformer Encoder.

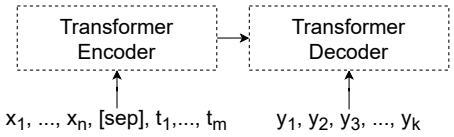

Fig. 4. Transformer model with source sentence and template concatenation via *[SEP]* token where x_i is source words, t_i is template tokens (words or POS tags), y_i is target words.

5 Experimental Results

Dataset. In order to prove the strength of our work, we conducted experiments on the four datasets: IWSLT 2014 German - English[6], IWSLT 2015 English - Vietnam, IWSLT 2017 English - French[7] and WMT 2014 English - German. The statistics of these data are shown in Table 1.

Table 1. Statistic information of NMT datasets.

Information	**IWSLT**			**WMT**
	de-en	**en-vi**	**en-fr**	**en-de**
# training examples	160K	133K	230K	4M
# development examples	7.3K	1.5K	1K	40K
# testing examples	6.8K	1.2K	1K	3K
# BPE operators	10K	10K	10K	40K

[6] test set is merged from dev2010, dev2012, tst2010, tst2011, tst2012
[7] test set identify is tst2015

Settings. To verify the performance of the proposed methods, we trained the following systems with the same settings: (1) the baseline NMT translation methods (TRANSFORMER) with 6 Self-Attention layers for the encoder and decoders; 8 heads for WMT14 dataset, and 4 heads for others; model size is 512 and hidden size is 2048; dropout is 0.3; (2) the NMT translation using both source template and target template where the template is generated from length-based (LENT) [15], our probability-based (PROBT), or our depth-based (DEPT) methods; the drop template threshold is selected in {0.5; 0.6; 0.7; 0.8; 0.9} similar to [15]. All datasets are pre-processed with the standard Moses toolkit[8]. We evaluated performance by averaging 5 latest checkpoints and compute BLEU score via SacreBleu[9] [7] on IWSLT 2017, WMT 2014 datasets and use multi-bleu script[10] on IWSLT 2014, 2015 datasets for comparable with previous published results.

Template Encoding Methods. To compare the effectiveness of TME and STC methods, we conducted experiments 5-10 using three types of generating target templates LENT, PROBT and DEPT on IWSLT 2014 dataset (Table 2). The TME method beats the STC method on all experiments of IWSLT 2014 de-en and en-de. With our observation, the reason is that TME method seems to have a *gating component* (Equation 4) that automatically select the useful information from template via cross-attention. The STC model used a simple *"[SEP]"* token to separate source sentence and template, and this model always utilizes these features for translating process while the TME model use learn-able parameters to adjust what information should be used.

Besides, to prove the effectiveness of *gating component* and *drop template mechanism*, we also conducted an ablation experiment on IWSLT 2014 de-en dataset (Table 3). Comparing to run 2 (Table 2) with runs 11, 12 (Table 3) and run 8 (Table 2) with runs 13, 14 (Table 3), we found that the performance of the NMT system is hurt a little bit, particularly in removing the *gating component*. These results are homologous to [15] conducted experiments about the *drop template mechanism*.

Template Types. Firstly, we consider the effectiveness of three types of templates: LENT, PROBT, DEPT. Our proposed DEPT template is generated as the highest abstract level representation of a sentence. Therefore, we argue that it contains useful structure information for the encoding process, especially on the source side. The evidence for this observation is shown in setting 10 (Table 2) with a stable improvement when compared to the competitive baseline Transformer model on all datasets. With setting 4 using DEPT on the target side, the result is just slightly improved in en-de datasets and decrease in others. We found that the quality of the prediction target template in the first phase of previous approach does not actually work well because it is tremendously challenging to predict the

[8] `https://github.com/moses-smt/mosesdecoder`

[9] `https://github.com/mjpost/sacrebleu`

[10] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

Table 2. Translation results for test sets of IWSLT 2014 German↔English, IWSLT 2015 English→Vietnam, IWSLT 2017 English→French and WMT 2014 English→German. The numbers in the pair of brackets are different values when compared to baseline model TRANSFORMER. The marked (*) result in WMT 2014 dataset refers to the run using Transformer-big setting [12] while the others use Transformer-base setting. The method ST-NMT by [15] is equal to setting 2 in our implementation.

Methods	**IWSLT14**		**IWSLT15**	**IWSLT17**	**WMT14**
	de-en	**en-de**	**en-vi**	**en-fr**	**en-de**
Previous works					
Transformer [12]	34.42	28.35	-	-	27.30
TreeTransformer [6]	35.96	29.47	-	-	28.40
BPE-dropout [8]	-	-	33.27	40.02	28.01
ST-NMT [15]	35.24	-	-	-	29.68*
Our implementation					
1. TRANSFORMER	35.93	29.63	32.20	39.37	27.27
Target side template					
2. TME +LENT	$36.07_{(+0.14)}$	$29.77_{(+0.14)}$	$31.81_{(-0.39)}$	$39.40_{(+0.03)}$	$27.07_{(-0.20)}$
3. TME +PROBT	$36.00_{(+0.07)}$	$29.72_{(+0.09)}$	$31.50_{(-0.70)}$	$38.97_{(-0.40)}$	$26.99_{(-0.28)}$
4. TME +DEPT	$36.04_{(+0.11)}$	$29.70_{(+0.07)}$	$31.73_{(-0.47)}$	$38.92_{(-0.45)}$	$27.09_{(-0.18)}$
Source side template					
5. STC +LENT	$35.79_{(-0.14)}$	$29.33_{(-0.30)}$	-	-	-
6. STC +PROBT	$35.85_{(-0.08)}$	$29.43_{(-0.20)}$	-	-	-
7. STC +DEPT	$35.84_{(-0.09)}$	$29.55_{(-0.08)}$	-	-	-
8. TME +LENT	$35.99_{(+0.06)}$	$29.69_{(+0.06)}$	$32.48_{(+0.28)}$	$39.11_{(-0.26)}$	$27.10_{(-0.17)}$
9. TME +PROBT	$36.04_{(+0.11)}$	$29.70_{(+0.07)}$	$\mathbf{32.50}_{(+0.30)}$	$\mathbf{39.56}_{(+0.19)}$	$\mathbf{27.34}_{(+0.07)}$
10. TME +DEPT	$\mathbf{36.19}_{(+0.26)}$	$\mathbf{29.80}_{(+0.17)}$	$32.36_{(+0.16)}$	$39.45_{(+0.08)}$	$27.16_{(-0.11)}$

Table 3. Translation results for ablation experiments removing *gating component* or *drop template mechanism* on test sets of IWSLT 2014 German→English.

Methods	**IWSLT14 de-en**
Target template	
11. TME +LENT -GATING	35.17
12. TME +LENT -DROP	35.77
Source template	
13. TME +LENT -GATING	35.67
14. TME +LENT -DROP	35.82

abstract representation of the target sentence based on the source sentence, and the output is usually repeated with some popular DepT templates. Differently, the LenT template proposed by [15] is more suitable in target side (settings 2, 8). The LenT and ProbT templates are the mixture of target words and POS tags, that is punched in a more detailed level than DepT. Although it is hard to predict the correct template in the first phase, the NMT systems have the advantage of the predicted words in the template for the final target sentence prediction.

Secondly, we consider the effect of template sides (source or target sides) on settings 2, 3, 4, and 8, 9, 10 described in Table 2. These results show that our proposed methods using templates on the source side are more effective than ones on the target side. Since the target templates need to be learned by an NMT model, the quality of the target templates prediction is lower than the source templates extraction, and the overall translation performance is decreased. In the IWSLT 2015 dataset, the Stanford Core-NLP tool does not support the Vietnamese language for syntactic parsing tasks, so we utilize a *spaCy* to parse constituent tree from a natural sentence. Therefore, the performance on the target side of this pair of languages drops sharply compared to the source side as well as other pairs of languages. Obviously, in this case, our proposed methods are more suitable and adaptive than the target-based templates for low-resource language (e.g. Vietnamese). The reason of this phenomena comes from the performance of syntactic parser in these kinds of languages and the error transmission in the original approach of [15]. The detailed comparison in Table 2 proves the strength of our model to deal with the low-resource language in machine translation. In the large-scaled dataset (WMT 2014), the template integration did not show clear improvement. We argue that the structure information of DepT template in large scale dataset is less meaningful due to the repetition in abstract template extracting from the syntactic shallow level.

Computation resource. Besides the performance, the other important aspects of the NMT system that affect the practicality are the training time and model size. With the approaches using target template, the NMT system has to contain two internal sub-modules which consists of one module to predict the target template, and another to predict the final sentence from source sentence and target template. Therefore, the computational time and model size is almost two times larger than our proposed approaches using a direct source template. Particularly, Table 4 shows the model size and training time of setting TME +DepT on IWSLT 2014 de-en dataset on both source and target sides for comparison.

Previous Works Comparison. Our method (TME +DepT) achieves the state-of-the-art result on IWSLT 2014 German→English with an 0.95-BLEU-score improvement when compare with [15]. This method also shows the improvement compared to the strong baselines in small datasets IWSLT 2014 English↔German, IWSLT 2015 English→Vietnam, and IWSLT 2017 English→French within the same settings. Comparing to the work of [6] on the WMT 2014 dataset, the TreeTransformer model can extract more structure information than methods

Table 4. Computation resource comparison between NMT system using a template on source side with target side. The values in the table present the number of learnable parameters (M = million) and training time in hours.

Method	Templ. generator	Target generator	Total
Source side	0M (0h)	59.5M (9.7h)	59.5M (9.7h)
Target side	37.1M (7.8h)	58.8M (9.3h)	95.9M (17.1h)

using a template because it encodes all the constituent trees instead of a particular depth level. Comparing to the SOTA result [9] on IWSLT 2015 and IWSLT 2017, our method can be incorporated with BPE-dropout technical to improve, however, we leave it for our future work.

6 Conclusion

This paper presents our proposed framework to integrate the syntactic template from source sentences into NMT models. Besides, we also propose two different kinds of template extraction methods to determine the abstract template of sentence. To prove the strength and robustness of our models, we also conduct the empirical experiments using either source or target side in the various generating methods for conventional Transformer models. With our detailed comparison and evaluation, our proposed architecture obtains the potential results against the original approach and competitive baselines in many benchmarks. Besides, we also analyze in detail the effect of the template on the translation process to accentuate the appropriate method for incorporating syntactic information into the encoding process.

References

1. L. Dong and M. Lapata. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia, July 2018. Association for Computational Linguistics.
2. A. Eriguchi, K. Hashimoto, and Y. Tsuruoka. Incorporating source-side phrase structures into neural machine translation. *Computational Linguistics*, 45(2):267–292, June 2019.
3. P. Gupta, J. Bigham, Y. Tsvetkov, and A. Pavel. Controlling dialogue generation with semantic exemplars. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3018–3029, Online, June 2021. Association for Computational Linguistics.
4. J. Hao, X. Wang, S. Shi, J. Zhang, and Z. Tu. Multi-granularity self-attention for neural machine translation. 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 887–897, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

5. P. M. Nguyen, K. Than, and M. L. Nguyen. Marking mechanism in sequence-to-sequence model for mapping language to logical form. *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–7, 2019.
6. X.-P. Nguyen, S. Joty, S. Hoi, and R. Socher. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*, 2020.
7. M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics.
8. I. Provilkov, D. Emelianenko, and E. Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online, July 2020. Association for Computational Linguistics.
9. R. Sennrich, B. Haddow, and A. Birch. 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, Aug. 2016. Association for Computational Linguistics.
10. W. Shang, C. Feng, T. Zhang, and D. Xu. Guiding neural machine translation with retrieved translation template. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2021.
11. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
12. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.
13. Y. Wang, H.-Y. Lee, and Y.-N. Chen. Tree transformer: Integrating tree structures into self-attention. 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 1061–1070, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.
14. H. Xu, J. van Genabith, D. Xiong, Q. Liu, and J. Zhang. Learning source phrase representations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 386–396, Online, July 2020. Association for Computational Linguistics.
15. J. Yang, S. Ma, D. Zhang, Z. Li, and M. Zhou. Improving neural machine translation with soft template prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5979–5989, Online, July 2020. Association for Computational Linguistics.
16. Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang. Sg-net: Syntax guided transformer for language representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.