ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240962

Diversity-Enhanced Learning for Unsupervised Syntactically Controlled Paraphrase Generation

Shaojuan Wu[a], Jitong Li[b], Yue Sun[c], Xiaowang Zhang[d,*] and Zhiyong Feng[e]

[a,b,c,d,e]College of Intelligence and Computing, Tianjin University
Tianjin Key Laboratory of Cognitive Computing and Application
Tianjin, China

Abstract. Syntactically controlled paraphrase generation is to generate diverse sentences that have the same semantics as the given original sentence but conform to the target syntactic structure. An optimal opportunity to enhance diversity is to make word substitutions during rephrasing based on syntactic control. Existing unsupervised methods have made great progress in syntactic control, but the generated paraphrases rarely have substitutions due to the limitation of training data. In this paper, we propose a Diversity syntactically controlled Paraphrase generation framework (DiPara), in which a novel training strategy is designed to obtain semantic sentences while using the given sentence as training objects. As diverse words vary the syntactic structure around them, we propose a phrase-aware attention mechanism to capture the syntactic structure associated with the current word. To achieve it, the linearized triple sequence is introduced to represent structure singly. Experiment results on two datasets show that DiPara outperforms strong baselines, especially diversity (Self-BLEU$_4$) is improved by 10.18% in ParaNMT-Small.

1 Introduction

Paraphrases are texts that convey the same meaning but in alternative vocabulary and syntactic structures [29, 1]. Syntactically Controlled Paraphrase Generation (SCPG) aims to produce diverse paraphrases of the given sentence by matching the specified target syntax [19, 21, 28]. Apart from the meaning and syntax of paraphrases, we also explicitly focus on the diversity of generated paraphrases because trivial paraphrases with minimal changes may not be helpful for applications [3]. It has been used in various language understanding tasks, such as creative generation [11, 20], adversarial example generation [7, 15], and question generation [17]. Existing syntactically controlled paraphrase generation networks [19, 26] have produced paraphrases with syntactic control, but they focus on large parallel paraphrase pairs for training. Unfortunately, paraphrase pairs are not only hardly accessible, but most of the established pairs are just rearrangements of words with different syntax [23].

To overcome an absence of parallel corpus, *Yang et al.*[24] first investigated the problem of unsupervised SCPG, which learns syntactically controlled paraphrase generation with non-parallel data, as shown in Figure 1. Since then, several unsupervised SCPG models have been reported in the literature and achieved competitive performance in both syntax control and semantic maintenance[5, 6]. However, our experiments have shown that existing unsupervised models perform poorly for the diversity of generated paraphrases (in Table 1). These methods still fail to produce diverse paraphrases, although they may alleviate the reliance on paraphrase pairs.

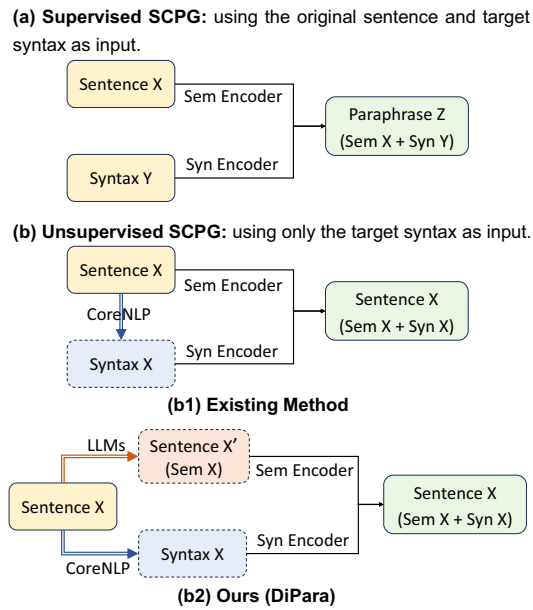

Figure 1. Differences between the existing methods and our method during SCPG (i.e., Syntactically Controlled Paraphrase Generation) training. 'sem' and 'syn' mean the semantics and syntax. The yellow and green background indicate the inputs and output of the model, respectively.

Furthermore, we construct a preliminary experiment to explore word diversity using Large Language Models (see Table 1), which have remarkable capabilities on semantic understanding[21, 26]. Surprisingly, the generated paraphrases are very diverse from both the original and target sentences. However, LLMs prefer to preserve the syntax of original sentences when generating diverse paraphrases in the SCPG task without being controlled by the target syntax at all, especially for compound sentences. This suggests that LLMs may have a negative impact on syntax control due to abundant linguistic knowledge. As a result, it is extremely challenging to attain both syntactic control and word diversity for unsupervised SCPG.

To address the above challenge, we propose a Diversity syntactically controlled Paraphrase generation framework (DiPara) that produces diverse paraphrases while conforming to target syntax. As

* Corresponding Author. Email: xiaowangzhang@tju.edu.cn

shown in Figure 1, we employ LLMs to generate multiple paraphrases with diverse words and determine the most appropriate semantic sentence by balancing semantics, syntax, and word. However, the involvement of diverse words changes the syntactic structure of their neighbors, and we observed that differences between two syntaxes are invariably reflected in the phrases. So, we propose phrase-aware attention to capture the structure associated with the current word. Motivated by this, the linearized triple sequence is designed to singly represent structures by splitting the content of the constituent parse tree before syntactic encoding.

In a nutshell, our contributions are as follows:

- We first present an LLM-based word diversity model to enhance the semantics of the original sentence by steadily producing diverse paraphrases performed with word substitutions.
- We propose a linearized triple sequence and phrase-aware attention mechanism to represent and capture the syntactic structure associated with the current word.
- We conduct extensive experiments with two datasets, and the results show that DiPara outperforms strong baselines in generating diverse paraphrases with target syntax. Moreover, the ablation study demonstrates the effectiveness of our proposed modules.

2 Related Work

SCPG aims to rewrite a text that conforms to the target syntax. More recent works typically utilize the Seq2Seq model [7] to generate diverse paraphrases by enhancing semantic encoder [26], syntactic encoder [25] or decoder [9, 26]. Particularly, some methods improve the quality of paraphrases by carefully selecting target syntactic structures [12, 28] and syntactic reordering [4, 19, 25]. These methods have made great advances in generating paraphrases with syntactic control, but they rely on large paraphrase pairs for training.

Considering paraphrase pairs are not easily available for many languages, [24] first proposes unsupervised SCPG, which does not require any parallel paraphrase data. Since then, [5] encodes the semantics without syntax by removing the position encoding. [6] employs abstract meaning representations to enhance semantic and syntactic embeddings further. Though these methods alleviate the reliance on paraphrase pairs, they still struggle to generate high-quality paraphrases.

In addition, large pre-trained models have been used for paraphrase generation. [3] present novelty-controlled paraphrase generation for different levels of novelty by specialized prompts. [21] propose a novel adaptation of prefix-tuning to reduce training costs.

In this work, we focus on the diversity of generated paraphrases and propose enhanced semantic encoding to capture subtle variations across words.

3 Approach

3.1 Problem Statement

Given a sentence $\boldsymbol{x_i} = \{x_i^1, x_i^2, \ldots, x_i^n\}$ and the target syntax s_i, Syntactically Controlled Paraphrase Generation (SCPG) is defined to generate a diverse paraphrase $\boldsymbol{p_i} = \{p_i^1, p_i^2, \ldots, p_i^m\}$ that conveys the same meaning of given sentence $\boldsymbol{x_i}$ while conforming to the target syntax s_i, where n and m are the length of given sentence and generated paraphrase, respectively.

For the unsupervised SCPG, the training set $D = \{x_i\}_{i=1}^{|D|}$ has only input sentence x_i. Therefore, the model requires reconstructing the sentence x_i using only the given sentence x_i and its syntax s_i', without annotated paraphrase pairs. As shown in Figure 2, the model aims to generate the same text as the input sentence "over the course of 6 years, we have lived in 15 cities.".

3.2 Enhanced Semantic Encoding

To facilitate diversity learning, we first promote LLM to obtain semantic sentences with the same semantic and diverse words as the original sentence for training. It assumes that LLMs can generate text with the same semantics and diverse words since they have been pre-trained on the large-scale corpus. Then, to ensure the quality of semantic sentences, we divide the process into two steps: semantic sentence generation and selection.

Semantic Sentence Generation. To exploit the potential of LLMs in generating diverse paraphrases, we first generate multiple candidate semantic sentences by constructing the instruction, consisting of the task description, a few demonstrations, and an original sentence.

Formally, given the task description of diverse semantic sentence generation I, we manually design k sentence pairs (x_1, y_1) with diverse words as demonstrations, formalized as $D_k = \{(x_1, y_1), (x_1, y_1), \ldots, (x_k, y_k)\}$. The original sentence x is also fed into LLMs, generating its corresponding semantic sentences y.

$$LLMs\left(I, D_k, x\right) = y$$

To ensure diversity, we highlight the diversity and quantity requirements in the task description. Manually designed sentence pairs are as diverse as possible while maintaining semantics.

Semantic Sentence Selection. To relieve the poor quality of paraphrases due to performance instability, we select the optimum semantic sentence by considering multiple metrics. Specifically, we first set the semantic threshold since the low self-BLEU value may be caused by word diversity or the wrong word. Then, they are ranked from calculated diversity and syntactic matching scores, respectively. We select the semantic sentence with high semantic and diversity scores but low syntax matching values. Low syntax matching reduces the syntactic impact during semantic encoding and increases the diversity of training samples.

In addition, the contextualized semantic embedding $\boldsymbol{z}_{sem}$ is obtained by feeding the semantic sentence y_i into the semantic encoder, formalized as:

$$\boldsymbol{z}_{sem} = Enc_{sem}\left(y_i^1, y_i^2, ..., y_i^{n'}\right) \tag{1}$$

where n' represents the length of sentence y_i.

3.3 Multi-level Syntactic Encoding

To capture the syntactic structure associated with the current word, we propose the multi-level syntactic encoding module, which consists of two stages: linearized triple sequence and syntax encoder.

Step 1: Linearized Triple Sequence. Following previous works [24], we use the constituency parse tree (without leaf nodes) to provide syntactic information obtained by the Stanford CoreNLP [13], as shown in Figure 2.

Given the original sentence x, we first obtain its constituency parse tree T_{syn} by the Stanford CoreNLP. Then, linearized triplet sequence is used to split it into content sequence Syn, structure sequences P_Syn and P_Parent, formalized as:

$$\begin{aligned} Syn &= \{n_i, i = 1, 2, \ldots, N\} \\ P\_\text{Syn} &= \{p_i, i = 1, 2, \ldots, N, p_i \in [1, N]\} \\ P\_\text{Parent} &= \{pa_i, i = 1, \ldots, N, pa_i \in [0, N - m]\} \end{aligned}$$

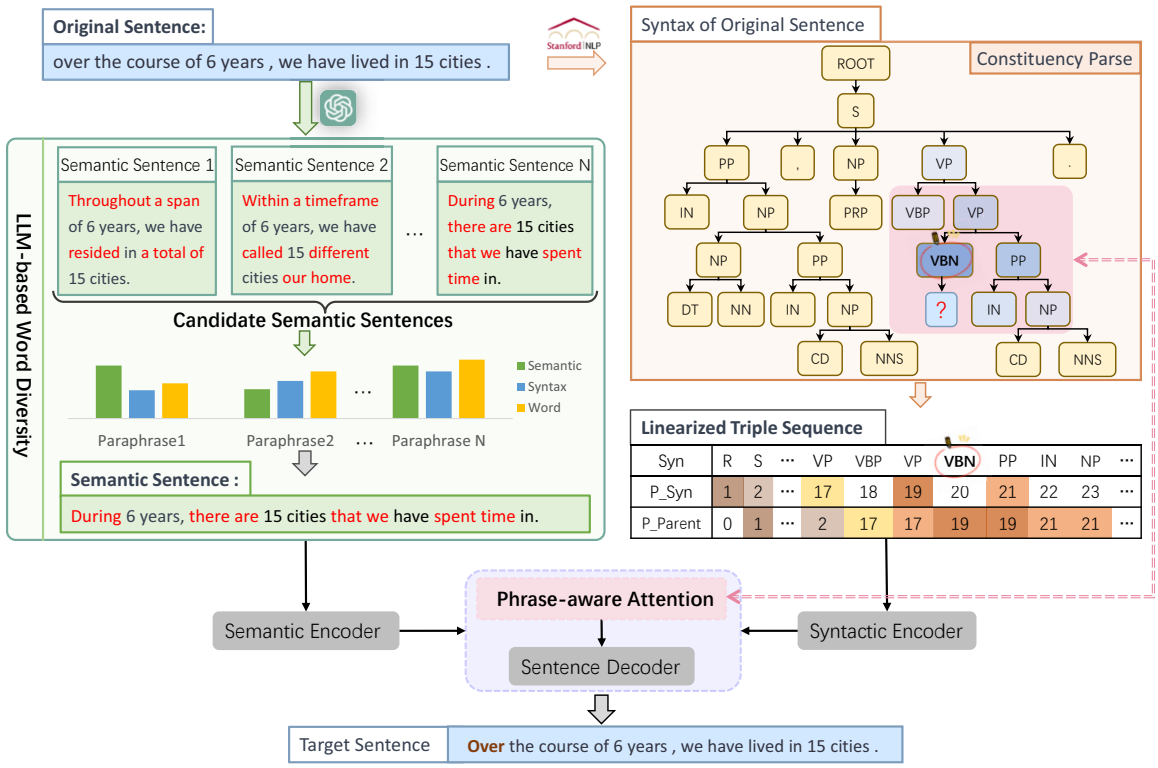

Figure 2. The overall architecture of our proposed method. It consists of an LLM-based word diversity module for semantic encoding, linearized triple sequences, and a phrase-aware attention mechanism for syntactic control.

where m is the number of POS tags and n_i is the syntactic node in T_{syn}. p_i and pa_i indicate the absolute position of each element and its parent node, which are encoded in a depth-first manner. Therefore, it satisfies that:

- If n_i is the parent node of n_j, then $p_i = pa_j$;
- If n_i and n_j are sibling nodes, then $pa_i = pa_j$.

Compared with the existing bracketed formats [7, 24], linearized triple sequence has the following advantages: Firstly, the constituency parse tree could be reconstructed more easily with $P_$Syn and $P_$Parent. Secondly, it provides structural information more directly through absolute positional coding. More importantly, it reduces the average length of sequences from 160 [10] to 80.

Step 2: Syntax Encoder. Considering that the attention range of syntactic nodes gradually expands as the number of layers, we employ a tree transformer to encode linearized triplet sequence.

For each node n_i, we first obtain the node embedding $\boldsymbol{n_i} \in \mathbb{R}^d$ and positional embedding $\boldsymbol{p_i} \in \mathbb{R}^d$, where d is the embedding dimension. The contextual matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is designed to focus on siblings and parent-child nodes, formalized as:

$$m_{ij} = \begin{cases} 1, & \text{if } pa_i = pa_j \text{ or } pa_{i(j)} = p_{j(i)}; \\ 0, & \text{otherwise} \end{cases}$$

At each layer, we compute the hidden state $\boldsymbol{h}_i$ of each node in a tree-structure manner.

$$\boldsymbol{h}_i^{enc} = Enc_{syn}(\boldsymbol{n_i} + \boldsymbol{p_i}, \boldsymbol{M_i})$$

Further, multi-head attention mechanism is utilized to get the contextual representation of the syntactic sequence. Finally, we obtain syntactic representation $\boldsymbol{z}_{syn}$ from the last layer of syntax encoder.

3.4 *Phrase-aware Attention*

Inspired by the observation that syntactic differences between two paraphrases are invariably reflected in the structure of phrases, we design a phrase-aware attention module to learn the importance distributions of syntactic nodes for each word adaptively.

Monotonic Attention. Since the Part-Of-Speech (POS) tagging of each word is deterministic and monotonic, we first obtain likelihood $\boldsymbol{l_t}$ that a syntactic node n_i would be the POS tag of the target word by computing the correlation r_t between syntactic representation $\boldsymbol{z}_{syn}$ and hidden states $\boldsymbol{h}_{t-1}^{dec}$.

$$\boldsymbol{r}_t = \boldsymbol{V}^T \tanh(\boldsymbol{W}_h^{mon}\boldsymbol{h}_{t-1}^{dec} + \boldsymbol{W}_{syn}^{mon}\boldsymbol{z}_{syn} + \boldsymbol{b}^{mon})$$

$$\boldsymbol{l}_t = \text{softmax}(\boldsymbol{r}_t + \epsilon)$$

where $\boldsymbol{V}, \boldsymbol{W}_h^{mon}, \boldsymbol{W}_{syn}^{mon}$ and $\boldsymbol{b}_{mon}$ are learnable weights. ϵ obeys the standard normal distribution.

Then, the importance distribution at the current moment $\boldsymbol{\alpha}_t$ is constrained by it at the former moment $\boldsymbol{\alpha}_{t-1}$, formalized as:

$$\boldsymbol{\alpha}_t = \boldsymbol{l}_t \cdot \text{Cprod}(1 - \boldsymbol{l}_t) \cdot \text{Csum}\left(\frac{\alpha_{t-1}}{\text{Cprod}(1-\boldsymbol{l}_t)}\right)$$

where $\text{Cprod}(\cdot)$ and $\text{Csum}(\cdot)$ are defined as:

$$\text{Cprod}(\boldsymbol{x}) = \left[1, x_1, x_1x_2, \ldots, \prod_{i=1}^{|x|-1} x_i\right]$$

$$\text{Csum}(\boldsymbol{x}) = \left[x_1, x_1 + x_2, \ldots, \sum_{i=1}^{|x|} x_i\right].$$

Cross-phrase Attention. After locating the POS tag of the target word, we learn l distance matrixes $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ to determine levels of other syntactic nodes centered on the POS tag. The element d_{ij}^l

means the probability that n_i and n_j belong to the l-level phrase, obtained as follows:

$$d_{ij}^{l} = c_{ij}^{l} - c_{ij}^{l-1}$$

where $d_{ij}^{1} = m_{ij}$, $l > 1$ and c_{ij}^{l} is computed as:

$$c_{ij}^{l} = \min\left(1, \sum_{k=1}^{N-1} c_{ik}^{l-1} \times m_{kj}\right)$$

Differently, d_{ij}^{l} indicates the distance between node i and node j is exactly equal to l, while c_{ij}^{l} indicates it is less than or equal to l. Based on this, the importance distribution of syntactic nodes at different levels is computed as follows:

$$\boldsymbol{\beta} = \sum_{i=1}^{l} \delta^{i} \times \boldsymbol{d}^{i}$$

where δ^{i} is trainable parameters.

Inter-phrase Attention. Considering the varying effects of syntactic nodes on the target word, even in the same phrase, we employ self-attention to capture semantic correlations between these nodes.

$$\boldsymbol{\gamma} = \mathrm{Softmax}\left(\frac{(\boldsymbol{W}_{q}^{in}\boldsymbol{z}_{syn})(\boldsymbol{W}_{k}^{in}\boldsymbol{z}_{syn})^{T}}{\sqrt{d}}\right)$$

where $\boldsymbol{W}_{q}^{in}$, and $\boldsymbol{W}_{k}^{in}$ are learnable weights.

Combining $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, it forms phrase-level attention vector $\boldsymbol{\eta} \in \mathbb{R}^{N\times N}$, formalized as:

$$\boldsymbol{\eta} = \boldsymbol{\alpha} \times (\boldsymbol{\beta} + \boldsymbol{\gamma}) \tag{2}$$

Finally, the syntactic structure associated with the target word $\boldsymbol{z}_{syn}^{t}$ is represented as:

$$\boldsymbol{z}_{syn}^{t} = \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{\eta}_{i,j}^{t} \cdot \boldsymbol{z}_{syn_j}$$

The final training objective of DiPara is to reconstruct the source sentence x by feeding the semantic embedding $\boldsymbol{z}_{sem}$ and syntactic embedding $\boldsymbol{z}_{syn}$ into the transformer decoder. Therefore, we minimize the following cross-entropy loss:

$$L = -\sum_{i=1}^{|D|} \log P(x_i | y, t, y_{1:t-1})$$

4 Experiments

4.1 Datasets

Following previous work [9], we evaluate DiPara on ParaNMT-Small and QQP-Pos.

- **ParaNMT-Small.** ParaNMT-Small [2] contains 500k paraphrase pairs for training, 500 and 800 manually labeled paraphrase pairs for validation and testing. It is a subset of the ParaNMT-50M dataset [23], constructed automatically by back-translating original English sentences. We produce 200k semantic enhanced paraphrase pairs during training and integrate them into the remaining data.
- **QQP-Pos** contains about 140K training pairs and 3K/3K pairs for testing/validation from the Quora Question Pairs (QQP) dataset [1]. Again, 7k enhanced paraphrase pairs are to be produced.

[1] https://www.kaggle.com/competitions/quora-question-pairs/

4.2 Evaluation Metrics

We evaluated three aspects using various evaluation metrics, including diversity, semantics, and syntax.

Diversity Metrics. We conducted the metric with words and phrases. In terms of words, we used **Self-BLEU**$_1$, i.e., BLEU-1 [14] between the input and generated paraphrase, to assess the capability of models in generating fresh words. **Self-BLEU**$_4$ [3] is calculated to account for n-gram overlaps. *Low Self-BLEU implies high diversity*.

Semantic Metrics. We employed **Reference-BLEU**$_4$ to evaluate the literal similarity between generated paraphrases and references. Further, we encoded the ground truth and generated paraphrase by **Sentence-BERT** [16] and then accessed their semantic similarity through cosine value.

Syntactic Metrics. We used the Exact Syntactic Match (**ESM**) and tree edit distance (**TED**) against the parse tree of the reference, following previous works[24, 28].

In addition, **iBLEU** [18] is calculated to evaluate the overall quality of paraphrases, calculated by iBLEU $=$ α Reference-BLEU$_4$ $-(1-\alpha)$ Self-BLEU$_4$, where α is set 0.8 following [28].

4.3 Baselines

We evaluate our method by comparing its performance with the following three kinds of models:

- To get a better sense of the natural diversity and semantic fidelity of the dataset, compared with the basic model: **Copying**, simply copying the original text; **Ground Truth**, using the ground truths as predictions themselves.
- To demonstrate the ability of syntactic control, compared with SCPG models: supervised methods, **SOW-REAP** [4], **AESOP** [19] and **SI-SCP** [25]. And unsupervised methods, including **SIVAE** [27], **SUP** [24] and **SynPG** [5].
- Methods based on LLMs: using GPT-3.5-Turbo as the base model: **ChatGPT (Zero-Shot)**, Give an original sentence and a target syntax, ChatGPT generate a paraphrase that is semantically consistent with the original sentence and conforms to the target syntax. **ChatGPT (Few-Shot)**, choosing three paraphrase pairs as demonstrations according to the corresponding formatting.

4.4 Implementation Details

All sentences in the datasets are parsed as constituency parse using Stanford CoreNLP [13]. We used the scheduled Adam optimizer [8] for optimization, and the learning rate was set to 2.0 for all experiments. We set the hidden state size to 300 (i.e., d), filter size to 1024, and head number to 4. The number of layers of the semantic encoder, syntax encoder, and sentence decoder were set to 4, 3, and 4, respectively. The batch size was set to 128. We used BPE tokens pretrained with 30000 iterations. All hyperparameter tuning was based on the BLEU score on the validation set.

During the process of evaluating diversity, we found that not only diversity is a factor of impact on the self-BLEU, but another possible factor is the generation of some irrelevant words. It seriously affects the authority of our evaluation. In addition, we first evaluate the semantic fidelity. Then, the top 30% paraphrases are selected to calculate the diversity metrics, and experimental results showed that these paraphrases are higher than 87 on Sentence-BERT for all SCGP models.

Table 1. Performance of syntactically controlled paraphrase generation. '**EP**' refers to "Enhanced Paraphrase pairs" generated by ChatGPT. '▷' is calculated from the trained model, publicly available in the original paper.

Model	Self-BLEU$_1$ (↓)	Self-BLEU$_4$ (↓)	Reference-BLEU$_4$ (↑)	i-BLEU(↑)	Sentence-BERT (↑)	ESM(↑)	TED(↓)
ParaNMT-Small							
Copying/Ground Truth	100/41.77	100 /9.96	9.96/100	-12.03/78.01	79.27/100	36.88/100	11.80/0
Supervised Methods							
SOW-REAP [4] ▷	65.03	24.89	27.00	16.62	67.77	-	-
AESOP [19] ▷	45.49	11.69	20.44	14.01	71.87	77.38	6.74
SI-SCP [25] ▷	46.23	13.02	27.81	19.64	76.92	88.87	5.70
Unsupervised Methods							
SIVAE [27]	-	20.90	12.80	6.06	70.80	82.60	-
SUP [24]	-	20.70	33.10	22.34	74.70	89.20	-
SynPG [5]	-	18.84	32.20	21.99	76.49	88.37	-
DiPara (w/o EP)	42.21	10.83	30.51	22.24	77.30	92.13	5.54
ChatGPT (Zero-shot)	40.24	9.18	10.56	6.61	77.98	42.50	13.76
ChatGPT (Few-shot)	44.27	21.12	13.78	6.80	**79.04**	43.75	11.12
DiPara (Ours)	**37.26**	**8.66**	**33.51**	**25.08**	78.11	**92.96**	**5.23**
QQP-Pos							
Copying/Ground Truth	100/42.76	100/14.25	14.25/100	-8.6/77.15	84.07/100	37.30/100	14.00/0
Supervised Methods							
SOW-REAP [4] ▷	66.19	25.78	36.55	24.08	66.13	-	-
AESOP [19] ▷	62.05	39.84	43.41	26.76	83.89	80.86	5.35
SI-SCP [25] ▷	45.57	19.10	48.83	35.24	88.11	81.43	5.20
Unsupervised Methods							
SIVAE [27]	-	29.00	32.60	20.28	76.00	81.7	-
SUP [24]	-	32.70	43.70	28.42	80.90	87.50	-
SynPG [5]	-	19.15	33.20	22.73	73.84	81.50	-
DiPara (w/o EP)	42.05	14.78	44.55	32.68	87.53	85.86	4.98
ChatGPT (Zero-shot)	47.31	17.39	11.18	5.47	89.20	34.62	17.93
ChatGPT (Few-shot)	46.59	20.59	12.23	5.67	**95.01**	29.13	15.61
DiPara (Ours)	**39.41**	**12.84**	**48.85**	**36.51**	88.37	**87.93**	**4.79**

4.5 Main Results

Table 1 summarizes the experimental results on ParaNMT-Small and QQP-Pos. We observe that DiPara achieves the best performance among all SCPG methods in terms of diversity and syntactic control without using parallel paraphrase pairs.

- DiPara achieves the best results on all three evaluation metrics of diversity, even compared with ChatGPT. It indicates that DiPara effectively generates diverse paraphrases by training enhanced paraphrase pairs with abundant word or phrase substitutions.
- For syntactic control, DiPara achieves the state-of-the-art ESM scores of 92.96 on ParaNMT-Small and 87.93 on QQP-Pos. In addition, it also improves 0.83 points and 2.07 points using enhanced paraphrase pairs. It indicates that diversity paraphrase pairs are also beneficial for improving syntactic control.
- In addition, DiPara is optimized in almost all the metrics on the semantic and is only weaker than ChatGPT on the Sentence-BERT metric. This suggests that the DiPara model can maintain semantics excellently during paraphrase generation.

In conclusion, DiPara greatly improved the performance of syntactically controlled paraphrase generation while balancing quality and diversity.

4.6 Human Evaluation

We further conduct the human evaluation on generated paraphrases, following previous work [7, 24, 28]. Specifically, we randomly sample 100 generated paraphrases from the ParaNMT test set. Three annotators are then asked to rate them from two aspects: the overall quality and diversity against the original sentence. For the overall quality, **0** means it is not a paraphrase at all, **1** means it is a paraphrase with some grammatical errors and **2** means it is a grammatically correct paraphrase. For the diversity, **0** means it is almost identical to the original sentence, **1** means it is a paraphrase with some new words, and **2** means it has a different syntax and words. We also let annotators evaluate syntactic controllability (ESM-H): the percentage of generated sentences that follow the given syntax.

Table 2 shows the results of human evaluation, which are somewhat consistent with the automatic metrics. DiPara is superior in producing diverse paraphrases with both new words and different syntaxes, which tend to follow the given target syntax.

Table 2. Human evaluation on ParaNMT dataset.

Model	**Quality** (↑)	**Diversity** (↑)	**ESM-H** (↑)
SynPG	1.01	0.73	89.0
ChatGPT	**1.89**	1.44	80.0
DiPara (Ours)	1.47	**1.53**	**96.0**

4.7 Ablation Study

To investigate the effectiveness of each module in the proposed method, we design several ablated versions of our model. The main differences between the variants and our proposed approach are displayed in Table 3.

The upper section of Table 3 shows the ablation study results on the test set in the ParaNMT dataset. From the table, we came to the following observations:

- As expected, *Baseline* gets the worst performance of all variants, and our method improves the base model by a large margin.
- Compared with the *Baseline*, *Baseline + Word Diversity* can obtain improved performances on three diversity metrics without a drop in semantic fidelity and syntactic control. The results show

Table 3. Ablation study on the ParaNMT.

Model	Self-BLEU$_1$(↓)	Self-BLEU$_4$(↓)	Reference-BLEU$_4$(↑)	iBLEU(↑)	Sentence-BERT	ESM(↑)	TED(↓)
Baseline	47.52	14.96	25.95	17.77	75.27	89.38	8.27
Baseline + Word Diversity	41.75	10.29	27.51	19.95	76.27	89.87	8.06
Baseline + Linearization	45.57	12.90	27.81	19.72	76.40	90.86	7.66
Baseline + Word Diversity + Linearization	38.80	9.27	30.31	22.28	77.23	90.75	6.57
Baseline + Word Diversity + Phrase-aware Attn	39.17	9.97	29.83	21.87	76.97	91.50	6.40
Baseline + Linearization + Phrase-aware Attn	42.21	10.83	31.91	23.36	77.30	92.13	5.94
DiPara (Ours)	**37.26**	8.66	**33.51**	**25.08**	**78.11**	92.96	**5.23**
w/o Monotonic Attention	37.72	8.75	32.04	23.88	77.60	**93.29**	5.38
w/o Cross-phrase Attention	38.24	9.04	33.03	24.62	77.92	93.21	5.67
w/o Inter-phrase Attention	37.40	**8.62**	32.71	24.44	78.09	92.01	5.85

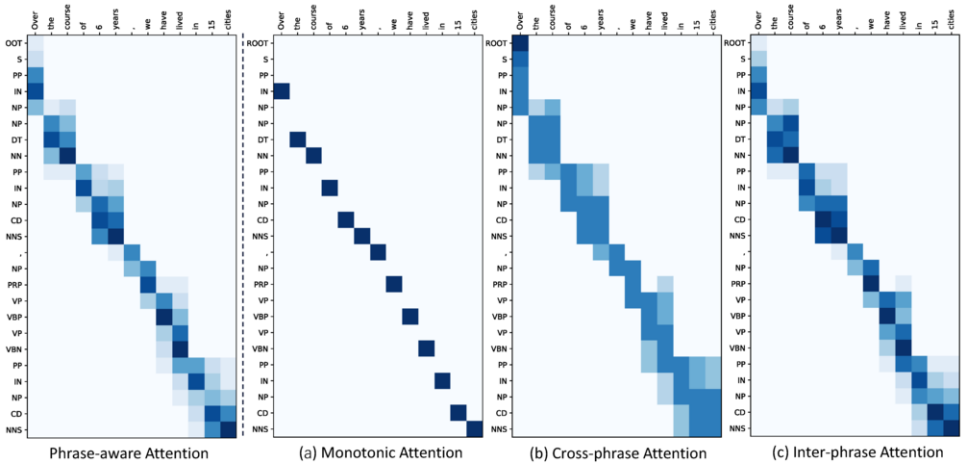

Figure 3. Attention scores of syntactic nodes for generating each words.

that ChatGPT-based augmented data helps generate high-quality paraphrased sentences with diversity.

- Compared with the *Baseline*, the performance of *Baseline + Linearization* is improved by 1.13 points and 1.48 points in Sentence-BERT and ESM, which indicates that combining the tree transformer encoder and the linearized triple sequence can capture richer syntactic structure information than the single-sequence processing approach.
- Moreover, a comparison between the *Baseline + Word Diversity / Baseline + Linearization* and the *Baseline + Word Diversity + Linearization* illustrates that jointly using word diversity and linearization can obtain a clear improvement on all metrics.
- We can observe that the *Baseline + Word Diversity + Phrase-aware Attn / Baseline + Linearization + Phrase-aware Attn* have further improvements to *Baseline + Word Diversity / Baseline + Linearization*, demonstrating the effectiveness of our phrase-aware attention.

Ablation study of phrase-level attention. We also conducted the ablation study to verify the necessity of three components of phrase-level attention. As shown in Table 3, the three phrase-level attentions collectively contribute to the model's performance in various aspects, demonstrating that the three components of the phrase-level attention are effective in capturing target syntactic structures.

5 Analysis

5.1 LLM-based Word Diversity Analysis

As shown in Table 1, enhanced paraphrase pairs are effective for improving the capability of generating diverse paraphrases. Specifically, removing EP severely decreases by 4.95 points and 2.17 points in terms of Self-BLEU$_1$ and Self-BLEU$_4$ on the ParaNMT-Small, respectively. Furthermore, as shown in Table 4, we compared differences between the original sentence and paraphrases to provide a visible look at the diversity. Paraphrases are from the ground truth of training set, and sentences are generated by baseline SCPG models, ChatGPT-based models and Dipara. It is obvious that paraphrases from the ChatGPT-based models have greater diversity than the the other baselines models but lack syntactic control, while Dipara achieves better diversity in satisfying the syntactic control.

In addition, we also conducted the ablation study (in the table 3), which effectively validates the effectiveness of LLM-based word diversity module.

5.2 Phrase-aware Attention Analysis

To have a clear view of the role that phrase-aware attention plays in DiPara, we visualize the attention scores of each syntactic node with respect to words in the sentence "over the course of 6 years, we have lived in 15 cities.", as shown in Figure 3. For the target word 'lived', the phrase-aware attention highlights 1-level syntactic nodes '*VP*', '*PP*' and even 2-level nodes '*VBP*', '*IN*', rather than just on its POS tag '*VBN*'. This aligns well with our design motivation, which adaptively captures the syntactic structure associated with the target word.

To further demonstrate the effectiveness of three components of phrase-level attention, we visualize the syntactic attention scores using only one attention mechanism. Specifically, monotonic attention enables the model to locate only the corresponding POS tag with each target word, as shown in Figure 3(a). It may be because POS tags are monotonic and deterministic, such as "have lived in" match

'*VBP*' '*VBN*' and '*IN*', respectively. Then, it is observed that the importance is increased for syntactic nodes, which are closer to the target word after using the cross-phrase component. Moreover, when at the same distance from the POS tag, they are mostly assigned same weight, such as '*VP* and *PP*' equally, '*VP*, *VBP*, *IN* and *NP*' also have the same attention value for the target word 'lived', as shown in Figure 3(b). It demonstrates that cross-phrase attention could effectively control syntactic structure in terms of levels. Furthermore, inter-phrase attention focused more on learning the importance of different syntactic nodes within the same level, as shown in Figure 3(c). For example, '*VP*, *VBP*, *IN* and *NP*' belong to the same level for the POS tag '*VBN*', but they are all calculated with different attention values. In addition, the performance is decreased after gradually removing three attention, which also verifies the necessity of three components (see Table 3).

5.3 *Qualitative Analysis*

We show a typical case on the ParaNMT-Small, which consists of the original sentence, target syntax and generated paraphrases by different models, as well as their corresponding constituency phrase. Moreover, models include baseline supervised SCPG models, ChatGPT-based models and DiPara, as shown in Table 4.

From an overall perspective, DiPara is able to balance diversity and syntactic control, though each model generated different results. Moreover, baseline SCPG models are good at syntactic control, while ChatGPT-based models are better at semantic restructuring.

Compared with the baseline SCPG models, our model not only generates a diverse paraphrase but also has excellent performance syntactic control. As shown in the last line of Table 4, DiPara generates the paraphrase "We have stayed in fifteen cities during six years.", different from the ground truth. But it is more diverse compared to the original sentence, while matching the target syntax exactly. It is challenging to generate diverse paraphrases for the baseline model. For example, the paraphrase "i lived in fifteen cities for six years." generated by AESOP has a near match in syntax. Unfortunately, there is only one keyword substitution, replacing 'we' with 'i', leading to semantics being broken.

ChatGPT-based models always generate somewhat diverse paraphrases while maintaining semantics. In addition, if the instruction excludes demonstration examples, it remains almost the syntax of the original sentence without being controlled by the target syntax at all, as shown in Table 4. However, if the instruction contains demonstration examples, the diversity of generated paraphrases decreases, even though the performance of syntactic control improves. For example, it generates the paraphrase "During a span of 6 years, we have resided in a total of 15 different cities." before demonstrations are added and generates "We have lived in 15 cities over the span of 6 years" afterward. Moreover, it has little effect on generating paraphrases whether demonstrations are added without inputting the target syntax.

In conclusion, DiPara can effectively generate diverse paraphrases conforming to the target syntax, which is attributed to the ability to balance semantics, syntax, and diversity.

6 Applications on Downstream Tasks

To further test the performance of DiPara in downstream tasks, we apply it to augment data for few-shot learning in text classification tasks. Specifically, we select SST-2, MRPC, and QQP classification tasks from GLUE [22] as evaluation benchmarks. Then, we randomly sample 500 instances from the training set and fine-tune roberta-base to obtain a baseline classifier. In addition, we utilize different paraphrase generation models to generate the paraphrases for the training set separately. The augmented data from the training set is used to train the classifier along with the original instances. We adopt the Accuracy metric to evaluate the model classification performance.

The results in Table 5 show that our method provides the greatest improvement compared to other methods. Specifically, the data augmentation of the DiPara model greatly improves the performance of the three classification tasks even before the training of enhanced paraphrase pairs. Meanwhile, ChatGPT's data augmentation method also achieved excellent results. Nevertheless, our DiPara model further improves the final performance after being enhanced with diverse, high-quality data. In conclusion, our DiPara performs best under all strategies, which shows that our approach can effectively enhance the application value of SCPG models in downstream tasks.

Table 4. An example of SCPG. Paraphrases are generated by baseline SCPG models, ChatGPT-based models and DiPara, with their constituency phrases on the right of the sentences. Blue fonts indicate the input. Magenta and grey fonts represent different words from the original sentence and different syntax from the target constituent phrase, respectively.

Models	Sentence	Constituency Phrase
Original Sentence	over the course of 6 years , we 've lived in 15 cities .	(ROOT (S (PP (IN) (NP (NP (DT) (NN)) (PP (IN) (NP (CD) (NNS))))) (,) (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (CD) (NNS))))) (.)))
Ground Truth	we have lived in fifteen cities over six years .	(ROOT (S (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (CD) (NNS)) (PP (IN) (NP (CD) (NNS)))))) (.)))
SOW-REAP	we 've lived in 15 cities over the course .	(ROOT (S (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (NP (CD) (NNS)) (PP (IN) (NP (DT) (NN)))))) (.)))
AESOP	i lived in fifteen cities for six years .	(ROOT (S (NP (PRP)) (VP (VBD) (PP (IN) (NP (CD) (NNS)) (PP (IN) (NP (CD) (NNS)))))) (.)))
SI-SCP	we 'v been living in 15 cities for six years .	(ROOT (S (NP (PRP)) (VP (VBP) (VP (VBN) (VP (VBG (PP (IN) (NP (CD) (NNS)) (PP (IN) (NP (CD) (NNS)))))))) (.)))
ChatGPT (Zero-Shot)	During a span of 6 years, we have resided in a total of 15 different cities.	(ROOT (S (PP (IN) (NP (DT) (NN) (PP (IN) (NP (CD) (NNS))))) (,) (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (NP (DT) (NN)) (PP (IN) (NP (CD) (JJ) (NNS)))))))) (.)))
ChatGPT (Few-Shot)	We have lived in 15 cities over the span of 6 years.	(ROOT (S (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (CD) (NNS)) (PP (IN) (NP (DT) (NN)) (PP (IN) (NP (CD) (NNS)))))))) (.)))
DiPara (Ours)	We have stayed in fifteen cities during six years.	(ROOT (S (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (CD) (NNS)) (PP (IN) (NP (CD) (NNS)))))) (.)))

Table 5. Performance of downstream tasks (i.e., MRPC, QQP, and SST-2) after adding paraphrases with different methods to the original baseline for data augmentation.

Methods	MRPC	QQP	SST-2
Baseline	80.44	68.38	67.83
+ ChatGPT	82.49	71.07	69.52
+ DiPara(w/o EP)	83.30	70.51	68.92
+ DiPara(Ours)	**86.69**	**74.06**	**70.33**

7 Conclusion

In this paper, we have presented DiPara, a novel framework that can effectively generate diverse paraphrases conforming to the target syntax by acquiring semantic sentences with diverse words and treating the given sentence as an objective. Experiments demonstrate that DiPara achieves the best performance in diversity and syntactic control across different datasets. We believe that DiPara opens up a new horizon for generating tasks (e.g., machine translation) that balance quality and diversity. It also provides an alternative to improve the diversity of enhanced data in many downstream tasks (e.g., question generation). In the future, we will consider merging SCPG models into large language models to enhance their generality and controllability by local fine-tuning.

References

[1] E. Bandel, R. Aharonov, M. Shmueli-Scheuer, I. Shnayderman, N. Slonim, and L. Ein-Dor. Quality controlled paraphrase generation. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 596–609, 2022. doi: 10.18653/v1/2022.acl-long.45.

[2] M. Chen, Q. Tang, S. Wiseman, and K. Gimpel. Controllable paraphrase generation with a syntactic exemplar. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 5972–5984, 2019. doi: 10.18653/v1/p19-1599.

[3] J. R. Chowdhury, Y. Zhuang, and S. Wang. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 10535–10544, 2022.

[4] T. Goyal and G. Durrett. Neural syntactic preordering for controlled paraphrase generation. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 238–252, 2020. doi: 10.18653/v1/2020.acl-main.22.

[5] K. Huang and K. Chang. Generating syntactically controlled paraphrases without using annotated parallel pairs. In P. Merlo, J. Tiedemann, and R. Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1022–1033, 2021. doi: 10.18653/v1/2021.eacl-main.88.

[6] K. Huang, V. Iyer, A. Kumar, S. Venkatapathy, K. Chang, and A. Galstyan. Unsupervised syntactically controlled paraphrase generation with abstract meaning representations. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing (EMNLP, Findings)*, pages 1547–1554, 2022.

[7] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In M. A. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 13th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1875–1885, 2018. doi: 10.18653/v1/n18-1170.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[9] A. Kumar, K. Ahuja, R. Vadapalli, and P. P. Talukdar. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:330–345, 2020. doi: 10.1162/tacl_a_00318.

[10] Y. Li, R. Feng, I. Rehg, and C. Zhang. Transformer-based neural text generation with syntactic guidance. *CoRR*, abs/2010.01737, 2020.

[11] K. Liu, J. Qiang, Y. Li, Y. Yuan, Y. Zhu, and K. Hua. Multilingual lexical simplification via paraphrase generation. In K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, and R. Radulescu, editors, *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pages 1529–1535. IOS Press, 2023. doi: 10.3233/FAIA230433.

[12] H. Luo, Y. Liu, P. Liu, and X. Liu. Vector-quantized prompt learning for paraphrase generation. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP*, pages 13389–13398, 2023.

[13] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL, System Demonstrations)*, pages 55–60, 2014. doi: 10.3115/v1/p14-5010.

[14] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002. doi: 10.3115/1073083.1073135.

[15] F. Qi, M. Li, Y. Chen, Z. Zhang, Z. Liu, Y. Wang, and M. Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, pages 443–453, 2021. doi: 10.18653/v1/2021.acl-long.37.

[16] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 23rd Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, 2019. doi: 10.18653/v1/D19-1410.

[17] A. Saxena, S. Chakrabarti, and P. P. Talukdar. Question answering over temporal knowledge graphs. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, pages 6663–6676, 2021. doi: 10.18653/V1/2021.ACL-LONG.520.

[18] H. Sun and M. Zhou. Joint learning of a dual SMT system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference: Short Papers*, pages 38–42, 2012.

[19] J. Sun, X. Ma, and N. Peng. AESOP: paraphrase generation with adaptive syntactic control. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5176–5189, 2021. doi: 10.18653/v1/2021.emnlp-main.420.

[20] Y. Tian, A. K. Sridhar, and N. Peng. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP, Finding)*, pages 1583–1593, 2021. doi: 10.18653/v1/2021.findings-emnlp.136.

[21] Y. Wan, K. Huang, and K. Chang. PIP: parse-instructed prefix for syntactically controlled paraphrase generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. doi: 10.48550/arXiv.2305.16701.

[22] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[23] J. Wieting and K. Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 451–462. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1042.

[24] E. Yang, M. Liu, D. Xiong, Y. Zhang, Y. Meng, C. Hu, J. Xu, and Y. Chen. Syntactically-informed unsupervised paraphrasing with nonparallel data. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2594–2604, 2021. doi: 10.18653/v1/2021.emnlp-main.203.

[25] E. Yang, C. Bai, D. Xiong, Y. Zhang, Y. Meng, J. Xu, and Y. Chen. Learning structural information for syntax-controlled paraphrase generation. In M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, editors, *Findings of the Association for Computational Linguistics: NAACL*, pages 2079–2090, 2022. doi: 10.18653/v1/2022.findings-naacl.160.

[26] E. Yang, M. Liu, D. Xiong, Y. Zhang, Y. Meng, J. Xu, and Y. Chen. Improving generation diversity via syntax-controlled paraphrasing. *Neurocomputing*, 485:103–113, 2022. doi: 10.1016/j.neucom.2022.02.020.

[27] X. Zhang, Y. Yang, S. Yuan, D. Shen, and L. Carin. Syntax-infused variational autoencoder for text generation. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 2069–2078, 2019. doi: 10.18653/V1/P19-1199.

[28] X. Zhang, S. Zhang, Y. Liang, Y. Chen, J. Liu, W. Han, and J. Xu. A quality-based syntactic template retriever for syntactically-controlled paraphrase generation. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9736–9748, 2023.

[29] J. Zhou and S. Bhat. Paraphrase generation: A survey of the state of the art. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5075–5086, 2021. doi: 10.18653/v1/2021.emnlp-main.414.