# Diversity-enhanced Learning for Unsupervised Syntactically Controlled Paraphrase Generation

Anonymous ACL submission

## 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 as 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.

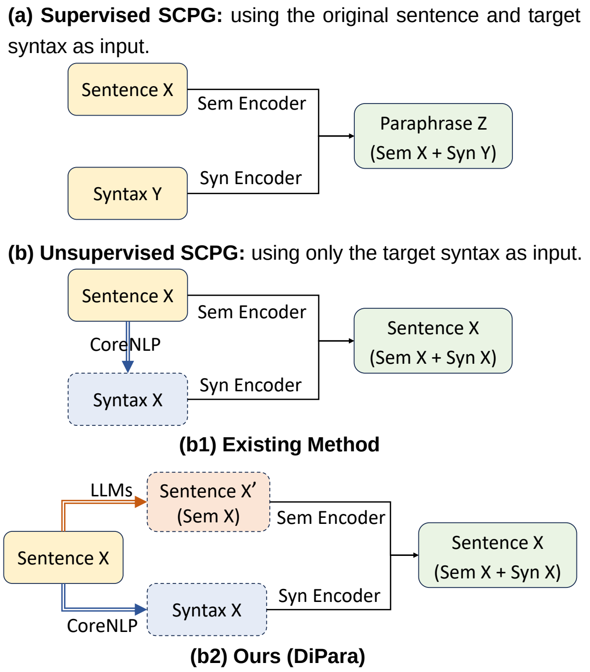


Figure 1: Difference between the supervised and unsupervised SCPG (i.e., Syntactically Controlled Paraphrase Generation) during training. 'sem' and 'syn' mean the semantics and syntax. The yellow and green ground indicate the inputs and output of the model, respectively.

## 1 Introduction

Paraphrases are texts that convey the same meaning but in alternative vocabulary and syntactic structures (Zhou and Bhat, 2021; Bandel et al., 2022). Syntactically Controlled Paraphrase Generation (SCPG) aims to produce diverse paraphrases of the given sentence by matching the specified target syntax (Sun et al., 2021; Wan et al., 2023; Zhang et al., 2023). It has been used in various language understanding tasks, such as creative generation (Tian et al., 2021), adversarial example generation (Iyyer et al., 2018; Qi et al., 2021), and question generation (Saxena et al., 2021). Unfortunately, paraphrase pairs are not easily available for many languages and are expensive to build (Wieting and Gimpel, 2018). *Yang et al.*(Yang et al., 2021a) 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(Huang and Chang, 2021; Huang et al., 2022).

However, our experiments have shown that existing unsupervised models perform poorly for the diversity of generated paraphrases (in Figure 1). Diverse paraphrasing is critical because trivial rephrasing with minimal changes may not be helpful for applications (Chowdhury et al., 2022). Furthermore, we construct a preliminary experiment to explore word diversity using Large Language Models (LLMs), which have remarkable capabilities on semantic understanding(Yang et al., 2022b; Wan et al., 2023). Surprisingly, the generated paraphrases are very diverse from both the original and target 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 shown in Figure 2, 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. 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 singly represent and capture the syntactic structure associated with the current word, respectively.
- 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 (Iyyer et al., 2018) to generate diverse paraphrases by enhancing semantic encoder (Yang et al., 2022b), syntactic encoder (Yang et al., 2022a) or decoder (Kumar et al., 2020; Yang et al., 2022b). Particularly, some methods improve the quality of paraphrases by carefully selecting target syntactic structures (Luo et al., 2023; Zhang et al., 2023) and syntactic reordering (Goyal and Durrett, 2020; Sun et al., 2021; Yang et al., 2022a). 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, (Yang et al., 2021b) first proposes unsupervised SCPG, which does not require any parallel paraphrase data. Since then, (Huang and Chang, 2021) encodes the semantics without syntax by removing the position encoding. (Huang et al., 2022) 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. (Chowdhury et al., 2022) present novelty-controlled paraphrase generation for different levels of novelty by specialized prompts. (Wan et al., 2023) 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 $\boldsymbol{x_i} = \{x_i^1, x_i^2, \dots, x_i^n\}$ and the target syntax $s_i$, Syntactically Controlled Paraphrase Generation (SCPG) is defined to generate a diverse paraphrase $\boldsymbol{p_i} = \{p_i^1, p_i^2, \dots, p_i^m\}$ 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

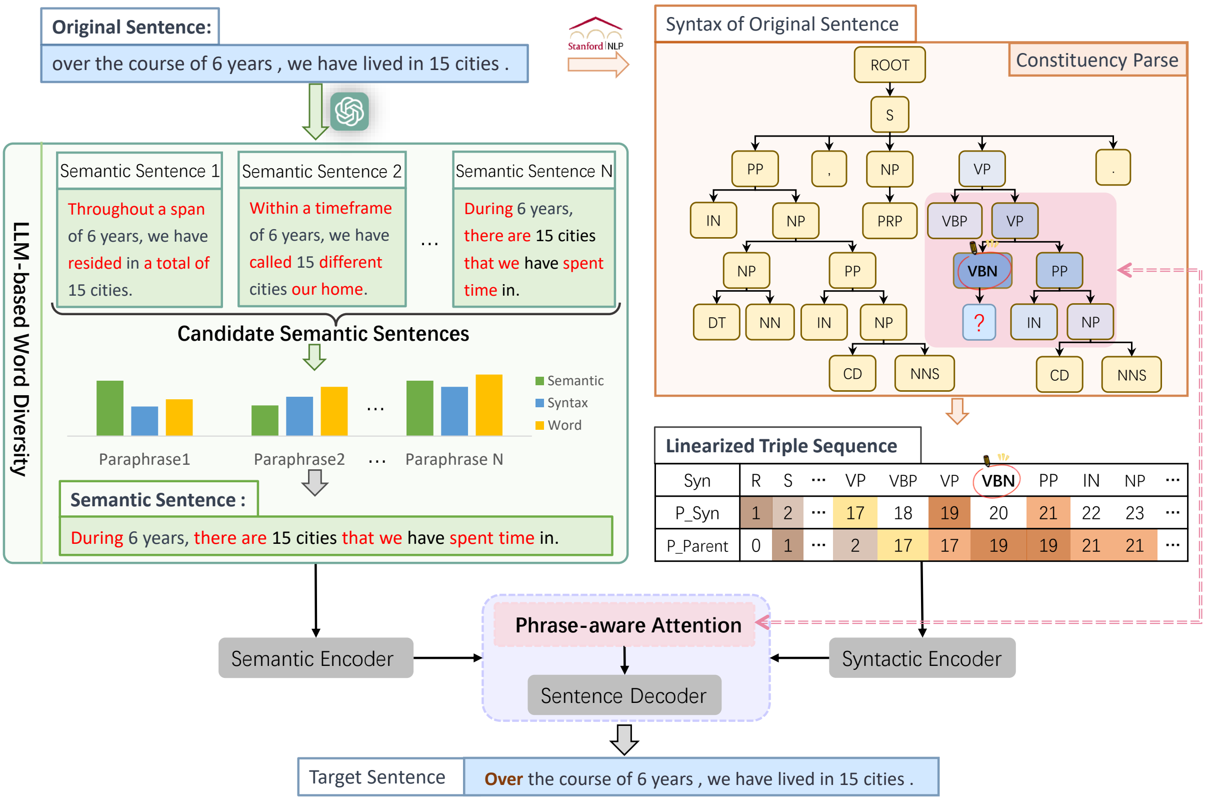

Original Sentence:
over the course of 6 years , we have lived in 15 cities .
LLM-based Word Diversity
Semantic Sentence 1
Throughout a span of 6 years, we have resided in a total of 15 cities.
Semantic Sentence 2
Within a timeframe of 6 years, we have called 15 different cities our home.
Semantic Sentence N
During 6 years, there are 15 cities that we have spent time in.
Candidate Semantic Sentences
Semantic
Syntax
Word
Paraphrase1
Paraphrase2
Paraphrase N
Semantic Sentence :
During 6 years, there are 15 cities that we have spent time in.
Stanford NLP
Syntax of Original Sentence
Constituency Parse
ROOT
S
PP
,
NP
VP
.
IN
NP
PRP
VBP
VP
NP
PP
VBN
PP
DT
NN
IN
NP
?
IN
NP
CD
NNS
CD
NNS
Linearized Triple Sequence
Syn R S ··· VP VBP VP VBN PP IN NP ···
P_Syn 1 2 ··· 17 18 19 20 21 22 23 ···
P_Parent 0 1 ··· 2 17 17 19 19 21 21 ···
Phrase-aware Attention
Semantic Encoder
Sentence Decoder
Syntactic Encoder
Target Sentence
Over the course of 6 years , we have lived in 15 cities .

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.

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 $D_k = \{(x_1, y_1), (x_1, y_1), \dots, (x_k, y_k)\}$. 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 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:

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

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 (Yang et al., 2021b), we use the constituency parse tree (without leaf nodes) to provide syntactic information obtained by the Stanford CoreNLP (Manning et al., 2014), 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 $P\_\text{Syn}$ and $P\_\text{Parent}$, formalized as:

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

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 (Iyyer et al., 2018; Yang et al., 2021b), linearized triple sequence has the following advantages: Firstly, the constituency parse tree could be reconstructed more easily with $P\_\text{Syn}$ and $P\_\text{Parent}$. Secondly, it provides structural information more directly through absolute positional coding. More importantly, it reduces the average length of sequences from 160 (Li et al., 2020) 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. $\epsilon$ 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:

$$
\begin{aligned}
\text{Cprod}(\boldsymbol{x}) &= \left[1, x_1, x_1 x_2, \ldots, \prod_{i=1}^{|x|-1} x_i\right] \\
\text{Csum}(\boldsymbol{x}) &= \left[x_1, x_1 + x_2, \ldots, \sum_{i=1}^{|x|} x_i\right].
\end{aligned}
$$

**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 $\delta^i$ 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.

$$\boldsymbol{\gamma} = \text{Softmax}\left(\frac{(\boldsymbol{W}_q^{in}\boldsymbol{z}_{syn})(\boldsymbol{W}_k^{in}\boldsymbol{z}_{syn})^T}{\sqrt{d}}\right)$$

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:

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

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:

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

## 4 Experiments

### 4.1 Datasets

Following previous work (Kumar et al., 2020), we evaluate DiPara on ParaNMT-Small and QQP-Pos.

- **ParaNMT-Small.** ParaNMT-Small (Chen et al., 2019) 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 (Wieting and Gimpel, 2018), 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.

### 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 (Papineni et al., 2002) between the input and generated paraphrase, to assess the capability of models in generating fresh words. **Self-BLEU**$_4$ (Chowdhury et al., 2022) 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** (Reimers and Gurevych, 2019) 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(Yang et al., 2021a; Zhang et al., 2023).

In addition, **iBLEU** (Sun and Zhou, 2012) is calculated to evaluate the overall quality of paraphrases, calculated by iBLEU $=$ $\alpha$ Reference-BLEU$_4$ $-(1-\alpha)$ Self-BLEU$_4$, where $\alpha$ is set 0.8 following (Zhang et al., 2023).

### 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, **Transformer** (Vaswani et al., 2017), **SOW-REAP** (Goyal and Durrett, 2020), **AESOP** (Sun et al., 2021) and **SI-SCP** (Yang et al., 2022a). And unsupervised methods, including **SIVAE** (Zhang et al., 2019), **SUP** (Yang et al., 2021a) and **SynPG** (Huang and Chang, 2021). Details of model descriptions are shown in Appendix B.

- Models based on ChatGPT: using GPT-3.5-Turbo as the model to generate paraphrases based on the combination of the original sentence and target syntax; **ChatGPT (Few-**

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

| 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 (Goyal and Durrett, 2020) ▷ | 65.03 | 24.89 | 27.00 | 16.62 | 67.77 | - | - |
| AESOP (Sun et al., 2021) ▷ | 45.49 | 11.69 | 20.44 | 14.01 | 71.87 | 77.38 | 6.74 |
| SI-SCP (Yang et al., 2022a) ▷ | 46.23 | 13.02 | 27.81 | 19.64 | 76.92 | 88.87 | 5.70 |
| *Unsupervised Methods* | | | | | | | |
| SIVAE (Zhang et al., 2019) | - | 20.90 | 12.80 | 6.06 | 70.80 | 82.60 | - |
| SUP (Yang et al., 2021a) | - | 20.70 | 33.10 | 22.34 | 74.70 | 89.20 | - |
| SynPG (Huang and Chang, 2021) | - | 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 (Goyal and Durrett, 2020) ▷ | 66.19 | 25.78 | 36.55 | 24.08 | 66.13 | - | - |
| AESOP (Sun et al., 2021) ▷ | 62.05 | 39.84 | 43.41 | 26.76 | 83.89 | 80.86 | 5.35 |
| SI-SCP (Yang et al., 2022a) ▷ | 45.57 | 19.10 | 48.83 | 35.24 | 88.11 | 81.43 | 5.20 |
| *Unsupervised Methods* | | | | | | | |
| SIVAE (Zhang et al., 2019) | - | 29.00 | 32.60 | 20.28 | 76.00 | 81.7 | - |
| SUP (Yang et al., 2021a) | - | 32.70 | 43.70 | 28.42 | 80.90 | 87.50 | - |
| SynPG (Huang and Chang, 2021) | - | 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** |

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.

**Shot)**, choosing three paraphrase pairs as demonstrations according to the corresponding formatting. Details of the instance formatting are shown in Appendix A (see Table 6).

### 4.4 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.

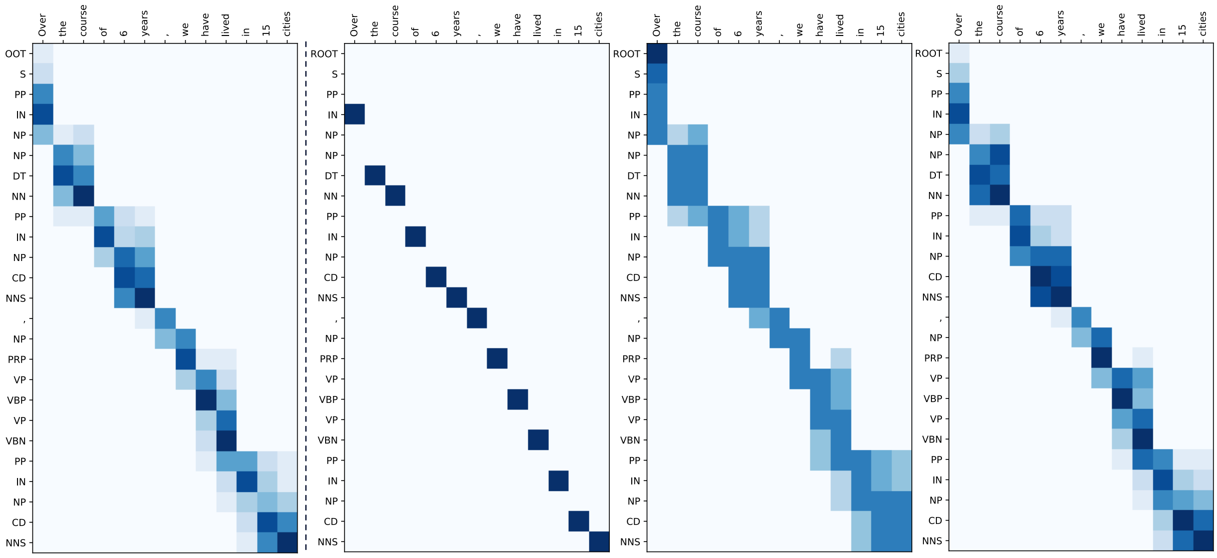
- For syntactic control, DiPara achieves the state-of-the-art ESM scores of 92.96 on ParaNMT-Small and 87.93 on QQP-Pos. In addition, it also improves 0.83 points and 2.07 points using enhanced paraphrase pairs. It indicates that diversity paraphrase pairs are also beneficial for improving syntactic control.

- In addition, DiPara is optimized in almost all the metrics on the semantic and is only weaker than the large language model 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.

### 4.5 Human Evaluation

We further conduct the human evaluation on generated paraphrases, following previous work (Iyyer et al., 2018; Yang et al., 2021b; Zhang et al., 2023). 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



Phrase-aware Attention (a) Monotonic Attention (b) Cross-phrase Attention (c) Inter-phrase Attention

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

| 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** |

Table 2: Human evaluation on ParaNMT dataset.

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. BiPare is superior in producing diverse paraphrases with both new words and different syntaxes, which tend to follow the given target syntax.

## 5 Analysis

In this section, we conduct fine-grained analysis regarding the improvements contributed by each module: LLM-based word diversity and phrase-aware attention.

### 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 3, 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 semantic sentences are generated by the the LLM-based word diversity module. It is obvious that semantic sentences have greater diversity than the ground truth. For example, the ground truth only has little new words (i.e.,, '*can*' and '*increase*'), but there are several diverse words in the semantic sentence, such as '*what*', '*best*', '*increase*', and so on. In addition, we also conducted the ablation study, which verified the effectiveness of LLM-based word diversity module. Details of experimental results and analysis are in Appendix D (see Table 7).

### 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

| Dataset | Original sentence | Ground Truth | Semantic Sentence |
|---|---|---|---|
| ParaNMT-Small | aren't you going to dress? | you're not going to dress? | Will you not attire yourself? |
| | alone and cut off from civilization, that's how mr.queen spent his last five years. | mr.queen spent his last 5 years alone, cut off from civilization. | For his final five years, Mr.Queen lived in isolation and removed from society. |
| QQP-Pos | how do i get more traffic on my website? | how can i increase the traffic on my website? | What is the best way to increase traffic on my website? |
| | how do you solve stoichiometry problems when given an excess product or reactant? | how do you solve a stoichiometry problem relating to excess reactants? | What is the method for solving stoichiometry problems when there is an excess of either product or reactant? |

Table 3: Lexical variability between semantic sentences generated by LLM-based Word Diversity and the ground truth for an original sentence in training sets.

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, detailed in Appendix D (see Table 7).

## 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 (Wang et al., 2019) as evaluation benchmarks. Then, we randomly sample 500 instances from the training set and fine-tune roberta-base(Liu et al., 2019) to obtain a baseline classifier as a few-shot baseline. 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 4 show that our method provides the greatest improvement to the baseline 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.

| Methods | MRPC | QQP | SST-2 |
|---|---|---|---|
| few-shot 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** |

Table 4: Performance of downstream tasks (i.e., MRPC, QQP, and SST-2) after adding paraphrases with different methods to the original baseline for data augmentation.

## 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.

## Limitations

We will discuss the limitations of our work from the following two aspects:

**Limited Paraphrase Pairs and Costs.** Since DiPara requires calling the API of large language model, it is potentially expensive compared to using other unsupervised PCPG models. Then, due to the budget limit, we only enhanced the diversity of a small portion of the dataset and evaluated it in English. However, it is interesting and meaningful to explore other languages as well, especially low-resource languages.

**Subject to Evaluation Metrics.** Diversity is rarely evaluated automatically because it is variable and ambiguous. Following previous work (Chowdhury et al., 2022), we also use Self-BLEU as evaluation metrics. However, experimental results show that high Self-BLEU means that generated paraphrases may be diverse or may be completely irrelevant. Therefore, we evaluated only parts of the data that ensure semantics. Nonetheless, it is imperative and meaningful to design an effective diversity metric.

Finally, we expect these limitations to be addressed in future work.

## References

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 596–609.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 5972–5984.

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 10535–10544.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 238–252.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1022–1033.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and Aram Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with abstract meaning representations. In *Proceedings of the 26th Conference on Empirical Methods in Natural Language Processing (EMNLP, Findings)*, pages 1547–1554.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 13th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1875–1885.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha P. Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:330–345.

Yinghao Li, Rui Feng, Isaac Rehg, and Chao Zhang. 2020. Transformer-based neural text generation with syntactic guidance. *CoRR*, abs/2010.01737.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Haotian Luo, Yixin Liu, Peidong Liu, and Xianggen Liu. 2023. Vector-quantized prompt learning for paraphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 13389–13398.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. 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.

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 (ACL)*, pages 311–318.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *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.

Apoorv Saxena, Soumen Chakrabarti, and Partha P. Talukdar. 2021. Question answering over temporal knowledge graphs. In *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.

Hong Sun and Ming Zhou. 2012. 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.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: paraphrase generation with adaptive syntactic control. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5176–5189.

Yufei Tian, Arvind Krishna Sridhar, and Nanyun Peng. 2021. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP, Finding)*, pages 1583–1593.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Yixin Wan, Kuan-Hao Huang, and Kai-Wei Chang. 2023. PIP: parse-instructed prefix for syntactically controlled paraphrase generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. 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.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 451–462. Association for Computational Linguistics.

Erguang Yang, Chenglin Bai, Deyi Xiong, Yujie Zhang, Yao Meng, Jinan Xu, and Yufeng Chen. 2022a. Learning structural information for syntax-controlled paraphrase generation. In *Findings of the Association for Computational Linguistics: NAACL*, pages 2079–2090.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2021a. Syntactically-informed unsupervised paraphrasing with non-parallel data. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2594–2604.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2021b. Syntactically-informed unsupervised paraphrasing with non-parallel data. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2594–2604.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Jinan Xu, and Yufeng Chen. 2022b. Improving generation diversity via syntax-controlled paraphrasing. *Neurocomputing*, 485:103–113.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019. Syntax-infused variational autoencoder for text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 2069–2078.

Xue Zhang, Songming Zhang, Yunlong Liang, Yufeng Chen, Jian Liu, Wenjuan Han, and Jinan Xu. 2023. A quality-based syntactic template retriever for syntactically-controlled paraphrase generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9736–9748.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5075–5086.

## A Large Language Models

In this section, we discuss the effect of prompting design on the SCPG task.

**Task Description.** Through preliminary experiments, we observe that the task description has a minimal effect on generating paraphrases in paraphrase generation and syntactically controlled paraphrase generation. The reason could be that LLMs have already developed a mature ability to generate paraphrases in the process of training with large-scale data. Therefore, LLMs already perform well even without adding specific task descriptions. However, both tasks still have their own focus, as shown in Table 5. Finally, we apply the task description by manual design to highlight the diversity of paraphrase generation.

**Instruction-formatted Design.** The quality of instruction instances has an important impact on the performance of the model. Therefore, we selected two potential methods to proceed with the formatted instance construction for comparison, including ChatGPT (Zero-Shot) and ChatGPT (Few-Shot). Details of the instruction-formatted instance are as shown in Table 6.

## B Baselines

We evaluate the ability of our method on diversity, semantic fidelity and syntactic control, compared with the following supervised SCPG methods:

- **Transformer.** (Vaswani et al., 2017), the syntactic encoder and semantic encoder both use the Transformer Encoder architecture and the decoder uses the Transformer Decoder architecture;
- **SOW-REAP** (Goyal and Durrett, 2020), a transformer-based encoder-decoder model, that uses syntactic rearrangement to enrich paraphrase variety while maintaining sentence quality;
- **AESOP** (Sun et al., 2021), a model that integrates pretrained language models with retrieval-based target syntactic parse selection module, that controls paraphrase generation with carefully chosen target syntactic structures;
- **SI-SCP** (Yang et al., 2022a), a model based on attention network, that designs a tree transformer to capture parent-child and sibling relation.

In addition, we also compared with unsupervised SCPG methods:

- **SIVAE** (Zhang et al., 2019), designing a syntax-infused variational autoencoder utilizing additional syntax information to improve the quality of sentence generation and paraphrase generation.
- **SUP** (Yang et al., 2021a), a model that presents a syntactically-informed unsupervised paraphrasing framework based on the conditional variational auto-encoder and uses the two-stage method to train the model.
- **SynPG** (Huang and Chang, 2021), treating the source sentence as a bag of words to decouple its semantics and syntax. Because its pre-trained model [2] was trained based on 21 million data, far more than ours, we retrained the model using our training dataset with all the parameters being set to the default value in the original papers.

## C Implementation Details

All sentences in the datasets are parsed as constituency parse using Stanford CoreNLP (Manning et al., 2014). We used the scheduled Adam optimizer (Kingma and Ba, 2015) 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 pre-trained 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.

## D Ablation Study

To investigate the effectiveness of each module in the proposed method, we design several ablated

[2] https://github.com/uclanlp/synpg

| Task | Task Descriptions |
|---|---|
| Paraphrase Generation | Given a sentence, please generate a paraphrase that has the **same content** but **different words as the given sentence** (Manually Design).<br>The task of paraphrase generation aims at **rephrasing a given text** while **retaining its meaning** (Chowdhury et al., 2022).<br>Paraphrase generation is a key technology of automatically generating a **restatement** for a given text (Yang et al., 2021a). |
| Syntactic Controlled Paraphrase Generation | Give an original sentence and a target syntax. Please generate a diverse paraphrase sentence that is **semantically consistent with the original sentence** and **conforms to the target syntax** (Manually Design).<br>Given an input sentence and a target syntax specification, an SCPG model aims to generate **paraphrases** that **satisfy the specific syntax requirement** (Wan et al., 2023).<br>Syntactically controlled paraphrase generation approaches aim to control **the format of generated paraphrases** by taking into account additional parse specifications as the inputs (Huang et al., 2022). |

Table 5: Task descriptions of paraphrase generation, syntactic controlled paraphrase generation. They are obtained from manual design and the definition of typical papers. Bolded words indicate key features of the task definition.

| Prompt ID | Prompt Template |
|---|---|
| ChatGPT (Zero-Shot) | **Task description**<br>Give an original sentence and a target syntax, please generate a paraphrase sentence that is semantically consistent with the original sentence and conforms to the target syntax.<br>**Input**<br>Original sentence: a huge black wolfish dog squatted down beside him .<br>Target syntax: (ROOT (SINV (PP (IN) (NP (PRP) (NN))) (VP (VBD)) (NP (DT) (JJ) (JJ) (JJ) (NN)) (.)))<br>**Output**<br>Please generate a paraphrase: |
| ChatGPT (Few-Shot) | **Task description**<br>Give an original sentence and a target syntax, please generate a paraphrase sentence that is semantically consistent with the original sentence and conforms to the target syntax.<br>**Demonstrations**<br>Original sentence: as shown by evidence , serious deficiencies exist in security systems .<br>Target syntax: (ROOT (S (NP (NP (NN)) (VP (VBG) (NP (EX)))) (VP (VBP) (NP (NP (JJ) (NNS)) (PP (IN) (NP (DT) (NN) (NNS))))) (.)))<br>Paraphrase sentence: Evidence confirming there are serious deficiencies in the security systems .<br>**Input**<br>Original sentence: a huge black wolfish dog squatted down beside him .<br>Target syntax: (ROOT (SINV (PP (IN) (NP (PRP) (NN))) (VP (VBD)) (NP (DT) (JJ) (JJ) (JJ) (NN)) (.)))<br>**Output**<br>Please generate a paraphrase: |

Table 6: An illustration of instance formatting and four different methods for constructing the instruction-formatted instances. The bolded font is just used to illustrate rather than as an input.

| 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 |

Table 7: Ablation study on the ParaNMT.

versions of our model. The main differences between the variants and our proposed approach are displayed in Table 7. Specifically,

- **Baseline.** The network removes our proposed modules, LLM-based word diversity, linearized triple sequence and phrase-aware attention. It consists of a semantic encoder, a syntactic encoder and a decoder.

- **Baseline + Word Diversity.** This variant adds the LLM-based word diversity module into the *Baseline*, which can generate augmented paraphrases with diverse words to enhance the semantics of the given sentence. By comparing it with the *Baseline*, we can evaluate the effectiveness of augmenting paraphrases with ChatGPT.

- **Baseline + Linearization.** This variant adds the linearized triple sequence module into the *Baseline*, which is used to separate syntactic contents and structures to keep the integrity of the input syntax structure. By comparing it with the *Baseline*, we can evaluate the effect of linearized triple sequence.

- **Baseline + Word Diversity + Linearization.** This variant incorporates both LLM-based word diversity and linearized triple sequence modules into the *Baseline*. By comparing it with the *Baseline*, we can evaluate the overall effect of our multi-level syntactic encoding.

- **Baseline + Word Diversity + Phrase-aware Attn / Baseline + Linearization + Phrase-aware Attn.** These variants further add the phrase-aware attention module to the *Baseline + Word Diversity / Baseline + Linearization*, respectively. By comparing them with *Baseline + Word Diversity* and *Baseline + Linearization*, we can evaluate the effect of the phrase-aware attention module.

The upper section of Table 7 shows the ablation study results on the test set in the paraNMT dataset. From the table, we can conclude the following observations:

1) as expected, among all the variants, *Baseline* gets the worst performance, and our method improves the base model by a large margin.

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

3) 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.

4) 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.

5) 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.

- **w/o Monotonic Attention.** This variant removes the monotonic attention from phrase-aware attention. By comparing it with *DiPara*, we can explore the effectiveness of the monotonic attention mechanism to capture the target syntactic structure.

- **w/o Cross-phrase Attention.** This variant removes the cross-phrase attention from phrase-aware attention. By comparing it with *DiPara*, we can investigate the effect of the cross-phrase attention module for syntactic nodes at different levels.

- **w/o Inter-phrase Attention.** This variant removes the inter-phrase attention from phrase-aware attention. By comparing it with *DiPara*, we can investigate the learning ability of the inter-phrase attention module for different syntactic nodes within the same level.

As shown in Table 7, the result of the model w/o inter-phrase attention declined under both the Sentence-BERT and ESM, especially under ESM by 0.95 points, which is caused there is no distinction between different syntactic nodes within the same level phrase without this module. Compared with the model w/o inter-phrase attention, the performance of the model w/o cross-phrase attention decreased by 0.19 points under Sentence-BERT and the performance of the diversity dropped by 0.84, 0.42 points under Self-BLEU$_1$ and Self-BLEU$_4$ respectively. The result shows that ignoring the impact of different levels of syntactic structure on the target words leads to poorer performance of the models in terms of semantics and diversity. The performance of the model w/o monotonic attention dropped under both the Sentence-BERT and diversity, which indicates that the monotonic attention mechanism has an important impact on improving the semantics and diversity of the generated paraphrases.

## E Qualitative Analysis

We show a typical case on the ParaNMT-Small, which consists of the given 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 8.

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 8, 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 almost remains the syntax of the original sentence without being controlled by the target syntax at all, as shown in Table 8. 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.

| Models | Sentence | Constituency Phrase |
|---|---|---|
| Given 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 w. S | 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 w. S + Few-Shot ICL | 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))))))) (.))) |
| ChatGPT w/o S | Throughout a span of 6 years, we have resided in 15 different cities. | (ROOT (S (PP (IN) (NP (NP (DT) (NN)) (PP (IN) (NP (CD) (NNS))))) (,) (NP (PRP)) (VP (VBP) (VP (VBN) (PP (IN) (NP (CD) (JJ) (NNS)))))) (.))) |
| ChatGPT w/o S + Few-Shot ICL | During 6 years, there are 15 cities that we have spent time in. | (ROOT (S (PP (IN) (NP(CD) (NNS))) (,) (NP (EX )) (VP (VBP) (NP (CD) (NNS) (SBAR (WHNP (WDT)) (S (NP (PRP)) (VP (VBP) (VP (VBN) (NP (NN)) (PP (IN)))))))))(.))) |
| **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 8: 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.