A Quality-based Syntactic Template Retriever for Syntactically-controlled Paraphrase Generation

Xue Zhang, Songming Zhang, Yunlong Liang, Yufeng Chen, Jian Liu, Wenjuan Han, Jinan Xu
Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
{xue_zhang,smzhang22,yunlongliang,chenyf,jianliu,wjhan,jaxu}@bjtu.edu.cn

Abstract

Existing syntactically-controlled paraphrase generation (SPG) models perform promisingly with human-annotated or well-chosen syntactic templates. However, the difficulty of obtaining such templates actually hinders the practical application of SPG models. For one thing, the prohibitive cost makes it unfeasible to manually design decent templates for every source sentence. For another, the templates automatically retrieved by current heuristic methods are usually unreliable for SPG models to generate qualified paraphrases. To escape this dilemma, we propose a novel Quality-based Syntactic Template Retriever (QSTR) to retrieve templates based on the quality of the to-be-generated paraphrases. Furthermore, for situations requiring multiple paraphrases for each source sentence, we design a Diverse Templates Search (DTS) algorithm, which can enhance the diversity between paraphrases without sacrificing quality. Experiments demonstrate that QSTR can significantly surpass existing retrieval methods in generating high-quality paraphrases and even perform comparably with human-annotated templates in terms of reference-free metrics. Additionally, human evaluation and the performance on downstream tasks using our generated paraphrases for data augmentation showcase the potential of our QSTR and DTS algorithm in practical scenarios.

1 Introduction

Paraphrase generation (PG) (Madnani and Dorr, 2010) is to rephrase a sentence into an alternative expression with the same semantics, which has been applied to many downstream tasks, such as question answering (Gan and Ng, 2019) and dialogue systems (Jolly et al., 2020; Gao et al., 2020; Panda et al., 2021; Liang et al., 2019, 2021, 2022). On this basis, to improve the syntactic diversity of paraphrases, syntactically-controlled paraphrase generation (SPG) is proposed and attracts extensive attention in the research community (Iyyer et al., 2018; Chen et al., 2019a; Kumar et al., 2020; Sun et al., 2021; Hosking and Lapata, 2021; Yang et al., 2021b, 2022a; Huang et al., 2022; Hosking et al., 2022). Different from traditional PG models, SPG models take syntactic templates as additional conditions to generate paraphrases conformed with the corresponding syntactic structures, whose forms generally include syntax parse trees and sentence exemplars. After years of research, SPG models can already generate syntax-conforming and high-quality paraphrases with human-annotated or well-selected syntactic templates (refer to the first case in Figure 1).

However, such promising performance heavily relies on those satisfying templates, whose difficult acquisition in practice largely hinders the applica-
tion of SPG models. Firstly, manually tailoring templates for every source sentence is practically unfeasible since it is time-consuming and laborious. Alternatively, automatically retrieving decent templates is also difficult, and unsuitable templates would induce semantic deviation or syntactic errors in the generated paraphrases (refer to the second case in Figure 1). On templates retrieval, current solutions are mostly heuristic methods and assume that suitable templates should satisfy certain conditions, e.g., high frequency in the corpus (Iyyer et al., 2018) or high syntactic similarity on the source side (Sun et al., 2021). The only exception is Yang et al. (2022a), which utilizes contrastive learning to train a retriever. Nonetheless, it also assumes that a suitable template should be well-aligned with the corresponding source sentence. Albeit plausible, the retrieval standards in these methods have no guarantee of the quality of the generated paraphrases, which may lead to the unstable performance of SPG models in practice.

To address this limitation, we propose a novel Quality-based Syntactic Template Retriever (QSTR) to retrieve templates that can directly improve the quality of the paraphrases generated by SPG models. Different from previous methods, given a source sentence and a candidate template, QSTR scores the template by estimating the quality of the to-be-generated paraphrase beforehand. To achieve this, we train QSTR by aligning its output score with the real quality score of the generated paraphrase based on a paraphrase-quality-based metric, i.e., ParaScore (Shen et al., 2022). With sufficient alignment, templates with higher scores from QSTR would be more probable to produce high-quality paraphrases. Moreover, when generating multiple paraphrases for each source sentence, we observe a common problem in QSTR and previous methods that the top-retrieved templates tend to be similar, which may result in similar or even repeated paraphrases. Aiming at this, we further design a Diverse Templates Search (DTS) algorithm to enhance the diversity between multiple paraphrases by restricting maximum syntactic similarity between candidate templates.

Experiments on two benchmarks demonstrate that QSTR can retrieve better templates than previous methods, which help existing SPG models generate paraphrases with higher quality. Additionally, the automatic and human evaluation showcase that our DTS algorithm can significantly improve current retrieval methods on the diversity between multiple paraphrases and meanwhile maintain their high quality. In the end, using templates from QSTR to generate paraphrases for data augmentation achieves better results than previous retrieval methods in two text classification tasks, which further indicates the potential of QSTR in downstream applications.

In summary, the major contributions of this paper are as follows:

- We propose a novel Quality-based Syntactic Template Retriever, which can retrieve suitable syntactic templates to generate high-quality paraphrases.
- To reduce the repetition when retrieving multiple templates by current methods, we design a diverse templates search algorithm that can increase the mutual diversity between different paraphrases without quality loss.
- The automatic and human evaluation results demonstrate the superiority of our method, and the performance in data augmentation for downstream tasks further prove the application values of QSTR in practical scenarios.

2 Related Work

As the syntactically-controlled paraphrase generation task is proposed (Iyyer et al., 2018) and has received increasing attention, previous work mainly focuses on improving the performance of the generated paraphrases conforming to the corresponding human-annotated templates. More specifically, most of them modify the model structures to better leverage the syntactic information of templates (Kumar et al., 2020; Yang et al., 2021a, 2022a). Furthermore, Sun et al. (2021) and Yang et al. (2022b) generate paraphrases based on the pre-trained language models, e.g., BART (Lewis et al., 2020), ProphetNet (Qi et al., 2020), and yield better performance.

Compared to the concentration on SPG models, only rare methods focus on how to obtain templates for these SPG models in practice. Among them, Iyyer et al. (2018) and Huang and Chang (2021) directly use the most frequent templates in the corpus. Sun et al. (2021) select the syntax parse trees of the target sentences in the corpus.

The code is publicly available at: https://github.com/XZhang00/QSTR
whose source sentences are syntactically similar to the input source sentence. Yang et al. (2022a) retrieve candidate templates based on distance in the embedding space. However, the retrieval standards in these heuristic methods cannot guarantee the quality of generated paraphrases. On the contrary, our QSTR directly predicts the quality of the paraphrases to be generated with the template, and the retrieved templates have greater potential to generate high-quality paraphrases.

3 Methodology

In this section, we first introduce our QSTR (§3.1), which includes the model architecture (§3.1.1) and the training objective (§3.1.2). Then we design a diverse templates search algorithm to improve the mutual diversity between multiple paraphrases (§3.2).

3.1 QSTR: Quality-based Syntactic Template Retriever

3.1.1 Model Architecture

As shown in Figure 2, QSTR has a two-tower architecture that contains a sentence encoder and a syntactic template encoder. The two encoders are used to encode the source sentence $x$ and the syntactic template $t$ respectively. Formally, given a source sentence with $n$ tokens $x = \{x_1, x_2, \ldots, x_n\}$ and a template\footnote{For example, the template “(ROOT (S (VP (LS) (S (VP)))) ( . ))” is formalized as \{’(’, ’ROOT’, ’(’, ’S’, ’(’, ’S’, ’(’, ’, ’, ’), ’)’)\}.} with $m$ constituents $t = \{t_1, t_2, \ldots, t_m\}$, the sentence encoder $Enc_s$ embeds $x$ into sentence embeddings $h^s$ and the syntactic template encoder $Enc_t$ embeds $t$ into template embeddings $h^t$:

$$h^s = (h^s_1, h^s_2, \ldots, h^s_m) = Enc_s(x_1, x_2, \ldots, x_n),$$

$$h^t = (h^t_1, h^t_2, \ldots, h^t_m) = Enc_t(t_1, t_2, \ldots, t_m).$$

To further extract the semantic and syntactic features, we add two feed-forward networks $FFN_s$ and $FFN_t$ after the two encoders. Then we can obtain the final sentence embeddings $e^s = \{e^s_1, e^s_2, \ldots, e^s_n\}$ and the final template embeddings $e^t = \{e^t_1, e^t_2, \ldots, e^t_m\}$:

$$e^s = FFN_s(h^s),$$

$$e^t = FFN_t(h^t).$$

To model the mapping relationships between the tokens in a sentence and the constituents of a related template, we calculate the pairwise dot product between two kinds of embeddings and obtain the correlation matrix $C$:

$$C_{n \times m} = e^s \cdot (e^t)^T,$$

where $C_{ij}$ indicates the degree of correlation between the token $x_i$ and the syntactic constituent $t_j$. Then we take the maximum value of each row/column in $C$ as the weight for weighted averaging sentence/template embeddings and obtain their final representations $v^s$ and $v^t$:

$$v^s = \frac{1}{n} \sum_{i=1}^{n} (\max_{j=1}^{m}(C_{ij} \ast e^s_i)),\quad (4)$$

$$v^t = \frac{1}{m} \sum_{j=1}^{m} (\max_{i=1}^{n}(C_{ij} \ast e^t_j)).\quad (5)$$

In the end, we concatenate $v^s$ and $v^t$ and transform it to a scalar $s$ through a linear layer and a Sigmoid function:

$$s = \text{Sigmoid}(W \cdot [v^s; v^t]),$$

Figure 2: The model architecture and the training process of QSTR. QSTR models the relationship between the source sentence $x$ and the template $t$ and maps it into a score $s$, which denotes a quality estimation of the paraphrase to be generated. Then, the training objective is to align the score $s$ with the true quality score $q$ of the paraphrase $p$ based on the ParaScore metric.
where $s$ is a score that represents the matching degree between the source sentence and the syntactic template.

In a nutshell, given a sentence $x$ and a template $t$, the goal of QSTR is to output a quality estimation $s$ for the future paraphrase through modeling the interaction between $x$ and $t$:

$$s = \text{QSTR}(x, t).$$

(7)

3.1.2 Training Objective

During training, we aim to align the estimated score $s$ from QSTR with the real quality value of the paraphrase. Towards this end, for each $x$, we randomly sample some templates from the whole template library $T$ as the candidate template set $T_k$, which also includes the templates of $x$ and the reference $y$ as more competitive candidates. Then, given $T_k = \{t_1, t_2, \ldots, t_k\}$, we can obtain the prior estimations $S_k = \{s_1, s_2, \ldots, s_k\}$ for these templates from QSTR by Eq.(7). At the same time, we can use the SPG model to generate the corresponding paraphrases $P_k = \{p_1, p_2, \ldots, p_k\}$ based on $T_k$ and evaluate their real quality. To acquire the quality scores of these paraphrases, we select the reference-based metric ParaScore$_{ref}$ (Shen et al., 2022) that has the highest correlation with human evaluation. Formally, given a source sentence $x$ and its reference $y$, the quality value $q_i$ of the paraphrase $p_i$ can be calculated by:

$$q_i = \text{ParaScore}_{ref}(p_i, x, y),$$

(8)

where we can use $q_i$ to construct the quality set $Q_k = \{q_1, q_2, \ldots, q_k\}$.

Next, we use the Mean Square Error (MSE) loss to quickly align the prior predictions $S_k$ with the posterior quality $Q_k$ quantitatively:

$$L_{mse} = \text{MSE}(S_k, Q_k).$$

(9)

Moreover, to better learn the quality ranks among $T_k$, we also calculate a pairwise rank loss $L_{rank}$ for $S_k$ according to $Q_k$:

$$L_{rank} = \sum_{i<j} \max \left( (\delta_{ij}^s - \delta_{ij}^q) * \mathbb{1}_{\delta_{ij}^s < 0}, 0 \right),$$

(10)

where $\delta_{ij}^s = s_i - s_j$, $\delta_{ij}^q = q_i - q_j$, and $\mathbb{1}_{\delta_{ij}^q < 0} = 1$ when $\delta_{ij}^q < 0$ otherwise 0.

In the end, the overall training objective $L$ consists of the above two loss functions:

$$L = \lambda_1 L_{mse} + \lambda_2 L_{rank}.$$

(11)

Algorithm 1 Diverse Templates Search Algorithm

**Input:** input sentence $x$, the whole template library $T = \{t_1, t_2, \ldots, t_{|T|}\}$, the number of the retrieved templates $d$, the syntactic diversity threshold $\beta$

**Output:** diverse templates set $T_d$

1: Initialize $T_d = \emptyset$ as a min heap.
2: for each $i \in [1, |T|]$ do
3: $s_i = \text{QSTR}(x, t_i)$
4: if $|T_d| < d$ then
5: $T_d$.push($t_i$)
6: else if $\min \text{TED}(t_i, T_d[*]) > \beta$ and $s_i > s_{T_d[0]}$ then
7: $T_d$.pop($T_d[0]$)
8: $T_d$.push($t_i$)
9: else
10: continue
11: end if
12: $T_d$.heap_sort() by $s_{T_d[*]}$
13: end for
14: return $T_d$

3.2 Diverse Templates Search Algorithm

In practice, the top templates retrieved by existing retrieval methods may have similar features, e.g., syntactic structures, which may lead to repetitions when generating multiple paraphrases for one source sentence. To improve the mutual diversity between multiple paraphrases while maintaining their high quality, we design a general diverse templates search (DTS) algorithm as described in Algorithm 1, which can be equipped with existing retrieval methods.

Taking QSTR as an example, given an input sentence $x$, we traverse the whole template library $T = \{t_1, t_2, \ldots, t_{|T|}\}$ and calculate a score $s_i$ for each $t_i$ (lines 2~3). Then, we maintain a min heap $T_d$ in a size $d$ to collect the satisfactory templates $t_i$ from $T$, which have high scores and meanwhile diverse syntactic structures between each other (lines 4~12). To find these templates, we calculate the Tree Edit Distance (TED) (Zhang and Shasha, 1989) between the template $t_i$ and templates in $T_d$ and ensure that the minimum TED value is greater than a threshold $\beta$ before appending $t_i$ to $T_d$ (line 6). After one traversal, the heap $T_d$ will contain the final $d$ qualified templates for diversely rephrasing the source sentence $x$. 

9739
4 Experimental Settings

4.1 Datasets and Implementation Details

Datasets. Following previous work (Sun et al., 2021; Yang et al., 2022a), we conduct our experiments on ParaNMT-Small (Chen et al., 2019b) and QQP-Pos (Kumar et al., 2020). Specifically, ParaNMT-Small contains about 500K paraphrase pairs for training, and 1300 manually labeled (source sentence, exemplar sentence, reference sentence) triples, which are split into 800/500 for the test/dev set. And QQP-Pos contains about 140K/3K/3K pairs/triples/triples for the train/test/dev set. In the test/dev set of two datasets, the exemplars are human-annotated sentences with similar syntactic structures as the reference sentences but different semantics. The function of exemplars is to provide their syntax parse trees as templates that guide SPG models to generate paraphrases syntactically close to references.

SPG Models. In our experiments, we use two strong SPG models, i.e., AESOP (Sun et al., 2021) and SI-SCP (Yang et al., 2022a), to evaluate the effectiveness of our QSTR. To the best of our knowledge, AESOP has state-of-the-art performance among previous SPG models. Besides, SI-SCP includes a novel tree transformer to model parent-child and sibling relations in the syntax parse trees and also achieves competitive performance.

Implementation Details. We use the Stanford CoreNLP toolkit (Manning et al., 2014) to obtain the syntax parse trees of the sentences, and we truncate all parse trees by height 4 and linearise them following Yang et al. (2022a). Then, we build the template library using the parse trees from the training set of each dataset. The two encoders in QSTR are initialized with Roberta-base (Liu et al., 2019) for both datasets. We use the scheduled AdamW optimizer with a learning rate of 3e-5 during training. The max length of sentences and templates are 64 and 192 respectively. The batch size is set to 32, and the size $k$ of $T_{k}$ is set to 10. We train our QSTR for 10 and 20 epochs on ParaNMT-Small and QQP-Pos respectively. The coefficients $\lambda_{1}$ and $\lambda_{2}$ in Eq.(11) are all set to 1. And the threshold $\beta$ in the DTS algorithm is set to 0.2.

4.2 Contrast Methods

Here we introduce all the contrast methods for obtaining templates. Moreover, we also conduct paraphrase generation with Vicuna-13B (Chiang et al., 2023) as a strong baseline.

Ref-as-Template. The syntax parse tree of the reference sentence is used as the template, which can be regarded as the ideal template for the source sentence.

Exemplar-as-Template. The syntax parse tree of the exemplar sentence is used as the template, which can be seen as the human-annotated template.

Random Template. We randomly select one template from the template library for SPG.

Freq-R. Following Iyyer et al. (2018), we choose the most frequent template in the template library for SPG.

AESOP-R. Sun et al. (2021) select the parse tree of the target sentence whose corresponding source sentence has the most similar syntactic structure to the input sentence.

SISCP-R. Yang et al. (2022a) encode the sentences and the templates into the same space and retrieve syntactic templates based on the similarities between their representations.

Vicuna-13B. We use this large language model (LLM) (Chiang et al., 2023) for zero-shot paraphrase generation. Please refer to Appendix A for more details.

4.3 Evaluation Metrics

Semantic Metrics. We use BLEU-R (Papineni et al., 2002) to evaluate literal similarity between generated paraphrases and references. To further measure the semantic similarity, we use sentence transformer to encode the sentences into embeddings and then calculate the cosine similarity between the paraphrase and the source/reference sentence as cos-S/cos-R.

Syntactic Metric. Following Bandel et al. (2022), we calculate the Tree Edit Distance (TED) (Zhang and Shasha, 1989) between the syntax trees of the paraphrase and the template to reflect how
much the paraphrase is syntactically conformed with the template. In our experiments, we observe that lower TED values generally indicate that the templates are more suitable for the source sentence.

**Diversity Metric.** BLEU-S calculates the BLEU scores between the paraphrase and the source sentence, whose lower values generally represent the paraphrases are more literally diverse from the source sentences.

**Comprehensive Metrics.** Based on BLEU-S and BLEU-R, we calculate the iBLEU (Sun and Zhou, 2012) to measure the overall quality of paraphrases by \( \text{iBLEU} = \alpha \text{BLEU-R} - (1 - \alpha) \text{BLEU-S} \), where we set \( \alpha = 0.8 \) following Hosking et al. (2022). Additionally, ParaScore (Shen et al., 2022) is the state-of-the-art metric for paraphrase quality evaluation, which can comprehensively evaluate semantic consistency and expression diversity of paraphrases. Therefore, we also use the reference-based and reference-free versions of ParaScore, i.e., \( \text{ParaScore}_{\text{ref}} \) and \( \text{ParaScore}_{\text{free}} \), as more convincing metrics. Between them, \( \text{ParaScore}_{\text{free}} \) can better reflect the quality of paraphrases when reference sentences are unknown in practical scenarios.

## 5 Results

### 5.1 Main Results

Table 1 shows the performance of paraphrases with different templates on two datasets, using the AESOP model for SPG. The results based on the SISCP model are presented in Appendix B. Totally, several conclusions can be drawn from the results:

1. Under the reference-based metrics, i.e., BLEU-R, cos-R, ParaScore\(_{\text{ref}}\) and iBLEU, our QSTR significantly surpasses other baselines, which demonstrates that the templates retrieved by QSTR are closer to the ideal templates.

2. The results of the reference-free metrics (i.e., BLEU-S, cos-S and ParaScore\(_{\text{free}}\)) also verify the superiority of our QSTR compared to other methods in practical scenarios and QSTR perform fully comparably with the human-labeled exemplar sentences (Exemplar-as-Template).

3. QSTR also achieves a much lower TED value than other retrieval methods, which indicates that the templates from QSTR are more suitable for the source sentence and the generated paraphrases conform more with the templates.

### Table 1: Performance of paraphrases generated with different kinds of templates using the AESOP model for SPG.

<table>
<thead>
<tr>
<th>Templates</th>
<th>BLEU-S↑</th>
<th>BLEU-R↑</th>
<th>iBLEU↑</th>
<th>cos-S↑</th>
<th>cos-R↑</th>
<th>ParaScore(_{\text{ref}})↑</th>
<th>ParaScore(_{\text{free}})↑</th>
<th>TED↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ideal Templates</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref-as-Template</td>
<td>22.430</td>
<td>60.260</td>
<td>43.722</td>
<td>0.850</td>
<td>0.914</td>
<td>0.856</td>
<td>0.948</td>
<td>0.096</td>
</tr>
<tr>
<td>Exemplar-as-Template</td>
<td>21.060</td>
<td>47.270</td>
<td>33.604</td>
<td>0.836</td>
<td>0.881</td>
<td>0.848</td>
<td>0.909</td>
<td>0.132</td>
</tr>
<tr>
<td><strong>Available Templates in Practice</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Template</td>
<td>15.030</td>
<td>10.480</td>
<td>5.378</td>
<td>0.728</td>
<td>0.703</td>
<td>0.794</td>
<td>0.739</td>
<td>0.214</td>
</tr>
<tr>
<td>Freq-R (Iyyer et al., 2018)</td>
<td>17.210</td>
<td>16.050</td>
<td>9.398</td>
<td>0.814</td>
<td>0.789</td>
<td>0.821</td>
<td>0.787</td>
<td>0.193</td>
</tr>
<tr>
<td>AESOP-R (Sun et al., 2021)</td>
<td>24.430</td>
<td>13.230</td>
<td>5.696</td>
<td>0.789</td>
<td>0.742</td>
<td>0.813</td>
<td>0.769</td>
<td>0.236</td>
</tr>
<tr>
<td>SISCP-R (Yang et al., 2022a)</td>
<td>28.180</td>
<td>18.580</td>
<td>9.228</td>
<td>0.851</td>
<td>0.809</td>
<td>0.837</td>
<td>0.825</td>
<td>0.221</td>
</tr>
<tr>
<td>QSTR (ours)</td>
<td>20.260*</td>
<td>22.080*</td>
<td>13.612</td>
<td>0.834</td>
<td>0.820*</td>
<td>0.851*</td>
<td>0.828*</td>
<td>0.163*</td>
</tr>
</tbody>
</table>

**LLM-based paraphrase generation**

<table>
<thead>
<tr>
<th>Templates</th>
<th>BLEU-S↑</th>
<th>BLEU-R↑</th>
<th>iBLEU↑</th>
<th>cos-S↑</th>
<th>cos-R↑</th>
<th>ParaScore(_{\text{ref}})↑</th>
<th>ParaScore(_{\text{free}})↑</th>
<th>TED↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vicuna-13B (Chiang et al., 2023)</td>
<td>13.302</td>
<td>5.060</td>
<td>1.388</td>
<td>0.851</td>
<td>0.762</td>
<td>0.861</td>
<td>0.786</td>
<td>/</td>
</tr>
</tbody>
</table>

ParaNMT-small

<table>
<thead>
<tr>
<th>Templates</th>
<th>BLEU-S↑</th>
<th>BLEU-R↑</th>
<th>iBLEU↑</th>
<th>cos-S↑</th>
<th>cos-R↑</th>
<th>ParaScore(_{\text{ref}})↑</th>
<th>ParaScore(_{\text{free}})↑</th>
<th>TED↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vicuna-13B (Chiang et al., 2023)</td>
<td>10.564</td>
<td>5.279</td>
<td>2.110</td>
<td>0.731</td>
<td>0.666</td>
<td>0.861</td>
<td>0.724</td>
<td>/</td>
</tr>
</tbody>
</table>

9741
Although some methods can achieve lower BLEU-S scores than QSTR (e.g., “Random Templates”), the corresponding cos-S scores are also significantly inferior, which means the generated paraphrases with these templates have poor semantic consistency with the source sentences.

Furthermore, despite the unfair comparison, we also report the results of Vicuna-13B, which conducts zero-shot paraphrasing without the need for templates. Although using a much smaller SPG model, QSTR can yield the closest performance to Vicuna-13B among template retrieval methods. And the detailed analysis is shown in Appendix A. In conclusion, these results sufficiently demonstrate that our QSTR can provide more suitable templates to generate high-quality paraphrases.

### 5.2 Mutual Diversity of Multiple Paraphrases

In this section, we evaluate the effectiveness of our DTS algorithm when generating multiple paraphrases for each source sentence. On the evaluation metrics, we first calculate the sentence-level repetition rate (Rep-Rate) between the paraphrases generated with the top-10 retrieval templates. And we use mutual-BLEU (M-BLEU) to measure the literal similarity between multiple paraphrases, which averages the corpus-level BLEU scores between different paraphrases. Additionally, we also report the average ParaScore\textsubscript{free} scores of 10 paraphrases for quality evaluation.

Table 2 presents the results of our DTS algorithm on two retrieval methods when retrieving 10 templates. The results showcase that equipped with DTS, the values of Rep-Rate and M-BLEU are decreased significantly, which means the DTS algorithm can effectively improve the mutual diversity between multiple paraphrases. Moreover, the stable ParaScore\textsubscript{free} scores of these paraphrases prove that the DTS algorithm has little impact on the quality of the paraphrases. To sum up, by combining our QSTR with the DTS algorithm, the SPG models can generate multiple paraphrases with both high mutual diversity and high quality.

### 6 Analysis

#### 6.1 Ablation Study

To verify the effectiveness of the two training objectives for QSTR, we further conduct an ablation study. Specifically, we calculate the Pearson correlation coefficient (PCC) between the predicted scores from QSTR and the true quality scores from ParaScore\textsubscript{ref}. Table 3 presents the results of QSTR when \( L_{\text{mse}} \) or \( L_{\text{rank}} \) is removed during training, which show that the correlations decline obviously without \( L_{\text{rank}} \) or \( L_{\text{mse}} \). Thus, both objectives benefit the training of QSTR and they can be complementary to each other.

#### 6.2 Human Evaluation

We further conduct the human evaluation on the paraphrases from the three retrieval methods (SISCP-R, QSTR, QSTR+DTS). Specifically, we randomly select 50 source sentences from the QQP-Pos test set and generate 5 paraphrases for each sentence using the AESOP model with templates from the three retrieval methods. Next, we let three annotators score each paraphrase from two aspects, i.e., the overall quality (1 ~ 5) and the diversity against the source sentence (1 ~ 5). The detailed guidelines are listed in Appendix C. Besides, we also define a paraphrase can be accepted if its quality score \( \geq 4 \) and diversity score \( \geq 3 \) and it is unique among the 5 paraphrases. The final evaluation results in Table 4 show that the paraphrases generated with QSTR have better quality and diversity under human evaluation. Moreover, the DTS can further promote the quality and diversity of the paraphrases and largely improve the acceptance rate.
Table 5: Paraphrases generated by the AESOP model with templates retrieved by different methods. The paraphrases in red/blue/black represent that they are terrible/repeated/eligible.

<table>
<thead>
<tr>
<th>Retrieval Methods</th>
<th>MRPC</th>
<th>QQP</th>
<th>SST-2</th>
<th>SST-2</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few-shot baseline</td>
<td>70.750</td>
<td>70.500</td>
<td>86.546</td>
<td>86.546</td>
<td>87.864</td>
</tr>
<tr>
<td>+ Random templates</td>
<td>70.333</td>
<td>69.833</td>
<td>86.216</td>
<td>86.875</td>
<td>88.029</td>
</tr>
<tr>
<td>+ Freq</td>
<td>71.833</td>
<td>71.167</td>
<td>85.832</td>
<td>86.985</td>
<td>87.974</td>
</tr>
<tr>
<td>+ AESOP-R</td>
<td>71.500</td>
<td>72.000</td>
<td>87.040</td>
<td>87.095</td>
<td>88.138</td>
</tr>
<tr>
<td>+ SISCP-R</td>
<td>71.583</td>
<td>71.917</td>
<td>86.930</td>
<td>87.150</td>
<td>88.468</td>
</tr>
<tr>
<td>+ QSTR (ours)</td>
<td><strong>72.250</strong></td>
<td><strong>72.417</strong></td>
<td><strong>87.864</strong></td>
<td><strong>87.534</strong></td>
<td><strong>88.523</strong></td>
</tr>
</tbody>
</table>

Table 6: Test accuracies of downstream tasks (i.e., MRPC, QQP, and SST-2) after adding paraphrases with different templates respectively to the original baseline for data augmentation. “Train Aug” means generating paraphrases for the training samples as the training corpus. “Test Aug” represents generating paraphrases for the test samples and conducting majority voting for the final predictions. “Train & Test Aug” combines the aforementioned two strategies. And results with the same mark † or ‡ are from the same model.

**6.3 Case Study**

We list 10 generated paraphrases of different retrieval methods for the same source sentence in Table 5. Among them, random templates produce the most inferior paraphrases, which shows that current SPG models are very sensitive to different templates. With the templates retrieved by SISCP-R, the paraphrases may be similar to each other and also have some syntactic or semantic errors. In contrast, our QSTR performs better on the quality of the paraphrases and the DTS algorithm further improves the mutual diversity of paraphrases.

**6.4 Applications on Downstream Tasks**

To further test the performance of QSTR on 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., 2018) as evaluation benchmarks. Then, we randomly sample 200 instances from the train set to fine-tune bert-base-uncased (Devlin et al., 2019) to obtain the baseline classifier as the few-shot baseline. In addition, we utilize the AESOP model with templates from different retrieval methods to generate paraphrases for the train set and the test set respectively. Specifically, the augmented data for the train set are used to train classifiers together with the original instances. As for the test set, we evaluate the augmented data as additional results, getting the majority voting as the final results. Moreover, we combine the aforementioned two strategies as a further attempt. Please refer to Appendix D for more training details.

The results in Table 6 present that our method brings the highest improvement over the baseline
7 Conclusion

In this work, we propose a quality-based template retriever (QSTR) to retrieve decent templates for high-quality SPG. Moreover, we develop a diverse templates search (DTS) algorithm to reduce the repetitions in multiple paraphrases. Experiments show that the SPG models can generate better paraphrases with the templates retrieved by our QSTR than other retrieval methods and our DTS algorithm further increases the mutual diversity of the multiple paraphrases without any loss of quality. Furthermore, the results of the human evaluation and the downstream task also demonstrate that our QSTR and DTS algorithm can retrieve better templates and help SPG models perform more stably in practice.

Limitations

Although Parascore (Shen et al., 2022) has been the state-of-the-art metric for the quality evaluation of paraphrases, it is still far from perfect as the supervision signal for QSTR. We will explore better metrics for evaluating the quality of paraphrases to guide the training of QSTR in future work. Moreover, we only utilize Vicuna-13B for zero-shot paraphrase generation, which leads to an unfair comparison with other methods. In future work, we will try to finetune Vicuna-13B on the SPG task and verify the effectiveness of our method with this new backbone.

Acknowledgements

The research work described in this paper has been supported by the National Natural Science Foundation of China (NSFC) via grants 61976016 and 61976015 and the Talent Fund of Beijing Jiaotong University via grant 2023XKRC006. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References


A The experiment based on Vicuna-13B

Vicuna-13B (Chiang et al., 2023) is a large-scale model that trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. And preliminary evaluation using GPT-4 as a judge shows Vicuna-13B outperforms other models like LLaMA and Stanford Alpaca in more than 90% cases (Chiang et al., 2023). To explore its performance on paraphrase generation, We use the instruction “Please give ten paraphrases of the next sentence in English, the input sentence. These paraphrases should have the same meaning and diverse syntactic structures with the given sentence.” to obtain the generated paraphrases from Vicuna-13B. The evaluation results on automatic metrics (as shown in Table 1) represent that Vicuna-13B can generate competitive paraphrases with our QSTR. Additionally, through our observation of specific cases, Vicuna-13B tends to add more additional information or expands the original sentence length during generating paraphrases. Several cases generated from Vicuna-13B are shown in Table 7. The advantage of this performance is that the generated paraphrases are more diverse against the source sentence, while the disadvantages are that the sentences become more redundant and the syntactic structures are not uncontrollable.

B The Results of SI-SCP

Firstly, to show the applicability of QSTR, the QSTR model used for the SI-SCP model is the same one used for the AESOP model (QSTR-based-on-AESOP). It is trained using the paraphrases generated by the AESOP model, which may leave a gap from the paraphrases from the SI-SCP model. Thus, we also conduct the more effective approach, using the SI-SCP model to generate paraphrases during the training process (QSTR-based-on-SI-SCP). Table 8 shows the performance of paraphrases with different templates from QSTR and other baselines on both datasets based on the SI-SCP backbone. The results showcase that our QSTR exhibits significant superiority compared to other baselines, and the QSTR-based-on-SI-SCP has better performance on most metrics than QSTR-based-on-AESOP. However, the promising results with QSTR-based-on-AESOP indicate that our QSTR may be directly applied to other SPG models without retraining.
source sentence: why do some people like cats more than dogs?

1. What is the reason behind the preference of some individuals for cats over dogs?
2. What makes certain people have a greater fondness for cats compared to dogs?
3. Why are some people more inclined towards cats rather than dogs?
4. What drives some individuals to prefer cats over dogs?
5. What is the explanation for the fact that some people prefer cats to dogs?
6. Why do certain people have a stronger affinity for cats compared to dogs?
7. What is the reason for the disparity in preferences between people who prefer cats and those who prefer dogs?
8. What sets apart the individuals who prefer cats from those who prefer dogs?
9. Why do some people find cats more appealing than dogs?
10. What are the factors that contribute to the variance in preferences for cats and dogs among people?

source sentence: how can i send a private message through quora?

1. How do I privately message someone on Quora?
2. Can you tell me the process for sending a private message on Quora?
3. I’m trying to send a confidential message on Quora, but I’m not sure how. How can you help?
4. Is there a way to send a discreet message on Quora? If so, how would I go about doing it?
5. How do I go about sending a secure message on Quora?
6. I’m trying to send a message on Quora that only the recipient can see. How can I do that?
7. Can you explain the steps to send a private message on Quora?
8. I’m trying to send a message on Quora that won’t show up in someone’s feed. How can I do that?
9. How do I send a message on Quora that won’t be visible to anyone else?
10. Can you provide me with the procedures to send a private message on Quora?

Table 7: Paraphrases directly generated by Vicuna-13B.

C Guidelines for human evaluation

The overall quality evaluates paraphrases from the perspectives of grammar correctness and semantic consistency with the source sentence, and the larger the score, the higher the quality. The detailed guidelines are as follows.

- "5" means the paraphrase is fully grammatically correct and completely semantically consistent with the source sentence.
- "4" means the paraphrase has a slight grammatical error, but still maintains the correct semantics, and can also be considered a valuable paraphrase.
- "3" means the paraphrase has a slight grammatical error and a minor semantic deviation.
- "2" means the paraphrase has a serious grammatical error and a major semantic deviation, but is still a complete sentence.
- "1" means the paraphrase is not a complete sentence or is totally irrelevant to the source sentence.

The diversity evaluates whether the paraphrase has diverse expressions compared to the input sentence based on whether the words used are the same or whether the syntactic structure is the same. And the larger the score, the higher the diversity. The detailed guidelines are as follows.

- "5" means the paraphrase has a very different syntax from the source sentence and adopts many new words and phrases.
- "4" means the paraphrase has a new syntax against the source sentence and revises some original words.
- "3" means the paraphrase has a new syntax against the source sentence but still adopts original words or phrases.
- "2" means the paraphrase has the same expression as the source sentence but adopts a few new words or phrases.
- "1" means the paraphrase is almost identical to the source sentence.

D Training details of Downstream Tasks

Since MRPC does not provide the official dev set, we randomly sample 1200 instances from the official test set as the final testing set and use the rest instances as the dev set. QQP adopts the same settings. As for the SST-2, we directly use the
<table>
<thead>
<tr>
<th>Templates</th>
<th>BLEU-S↓</th>
<th>BLEU-R↑</th>
<th>iBLEU↑</th>
<th>cos-S↑</th>
<th>cos-R↑</th>
<th>ParaScore_free↑</th>
<th>ParaScore_ref↑</th>
<th>TED↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal Templates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref-as-Template</td>
<td>23.810</td>
<td>52.760</td>
<td>37.446</td>
<td>0.838</td>
<td>0.888</td>
<td>0.844</td>
<td>0.920</td>
<td>0.136</td>
</tr>
<tr>
<td>Exemplar-as-Template</td>
<td>23.160</td>
<td>42.730</td>
<td>29.552</td>
<td>0.829</td>
<td>0.860</td>
<td>0.840</td>
<td>0.888</td>
<td>0.181</td>
</tr>
<tr>
<td>Available Templates in Practice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Template</td>
<td>20.540</td>
<td>15.920</td>
<td>8.628</td>
<td>0.755</td>
<td>0.732</td>
<td>0.802</td>
<td>0.768</td>
<td>0.301</td>
</tr>
<tr>
<td>Freq-R (Iyyer et al., 2018)</td>
<td>22.120</td>
<td>18.190</td>
<td>10.128</td>
<td>0.795</td>
<td>0.776</td>
<td>0.811</td>
<td>0.790</td>
<td>0.208</td>
</tr>
<tr>
<td>AESOP-R (Sun et al., 2021)</td>
<td>28.040</td>
<td>17.190</td>
<td>8.144</td>
<td>0.790</td>
<td>0.761</td>
<td>0.806</td>
<td>0.790</td>
<td>0.323</td>
</tr>
<tr>
<td>SISCP-R (Yang et al., 2022a)</td>
<td>29.230</td>
<td>21.050</td>
<td>10.994</td>
<td>0.839</td>
<td>0.805</td>
<td>0.820</td>
<td>0.817</td>
<td>0.284</td>
</tr>
<tr>
<td>QSTR-based-on-AESOP (ours)</td>
<td>21.520</td>
<td>22.540</td>
<td>13.728</td>
<td>0.815</td>
<td>0.804</td>
<td>0.833</td>
<td>0.820</td>
<td>0.220</td>
</tr>
<tr>
<td>QSTR-based-on-SI-SCP (ours)</td>
<td>24.580</td>
<td>22.500</td>
<td>13.084</td>
<td>0.828</td>
<td>0.810</td>
<td>0.850</td>
<td>0.829</td>
<td>0.214</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Templates</th>
<th>BLEU-S↓</th>
<th>BLEU-R↑</th>
<th>iBLEU↑</th>
<th>cos-S↑</th>
<th>cos-R↑</th>
<th>ParaScore_free↑</th>
<th>ParaScore_ref↑</th>
<th>TED↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal Templates</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref-as-Template</td>
<td>12.060</td>
<td>26.130</td>
<td>18.492</td>
<td>0.711</td>
<td>0.758</td>
<td>0.837</td>
<td>0.839</td>
<td>0.158</td>
</tr>
<tr>
<td>Exemplar-as-Template</td>
<td>11.980</td>
<td>18.950</td>
<td>12.764</td>
<td>0.665</td>
<td>0.686</td>
<td>0.811</td>
<td>0.785</td>
<td>0.191</td>
</tr>
<tr>
<td>Available Templates in Practice</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Template</td>
<td>10.110</td>
<td>6.870</td>
<td>3.474</td>
<td>0.596</td>
<td>0.562</td>
<td>0.748</td>
<td>0.653</td>
<td>0.256</td>
</tr>
<tr>
<td>Freq-R (Iyyer et al., 2018)</td>
<td>14.030</td>
<td>9.520</td>
<td>4.810</td>
<td>0.643</td>
<td>0.607</td>
<td>0.786</td>
<td>0.700</td>
<td>0.312</td>
</tr>
<tr>
<td>AESOP-R (Sun et al., 2021)</td>
<td>13.340</td>
<td>9.470</td>
<td>4.908</td>
<td>0.664</td>
<td>0.615</td>
<td>0.793</td>
<td>0.711</td>
<td>0.231</td>
</tr>
<tr>
<td>SISCP-R (Yang et al., 2022a)</td>
<td>17.200</td>
<td>11.810</td>
<td>6.008</td>
<td>0.723</td>
<td>0.676</td>
<td>0.807</td>
<td>0.779</td>
<td>0.190</td>
</tr>
<tr>
<td>QSTR-based-on-AESOP (ours)</td>
<td>14.140</td>
<td>13.830</td>
<td>8.432</td>
<td>0.694</td>
<td>0.675</td>
<td>0.823</td>
<td>0.768</td>
<td>0.157</td>
</tr>
<tr>
<td>QSTR-based-on-SI-SCP (ours)</td>
<td>16.070</td>
<td>12.770</td>
<td>7.002</td>
<td>0.710</td>
<td>0.680</td>
<td>0.833</td>
<td>0.781</td>
<td>0.167</td>
</tr>
</tbody>
</table>

Table 8: Performance of paraphrases with different kinds of templates based on the SI-SCP backbone. Metrics with ↑ means the higher value is better, while ↓ means the lower value is better. And the **bold** and underline represent the best and the second best respectively. For all retrieval methods in “Available Templates in Practice”, we use the top-1 retrieved template for each source sentence to generate the paraphrase.

official test set and dev set. All classifiers are initialized with bert-base-uncased (Devlin et al., 2019). The batch size is 128, 64, and 64 for MRPC, QQP, and SST-2 respectively. The learning rate is $10^{-4}$ and the number of training epochs is 20. And we use AdamW optimizer with weight decay being $10^{-5}$. 

9748