Abstract

Neural text generation models are likely to suffer from the low-diversity problem. Various decoding strategies and training-based methods have been proposed to promote diversity only by exploiting contextual features, but rarely do they consider incorporating syntactic structure clues. In this work, we propose using linguistic annotation, i.e., part-of-speech (POS), to guide the text generation. In detail, we introduce POS Guided Softmax to explicitly model two posterior probabilities: (i) next-POS, and (ii) next-token from the vocabulary of the target POS. A POS Guided Sampling strategy is further proposed to address the low-diversity problem by enriching the diversity of POS. Extensive experiments and human evaluations demonstrate that, compared with existing state-of-the-art methods, our POS Guided Softmax and Sampling (POSG) can generate more diverse text while maintaining comparable quality.

1 Introduction

Maximum likelihood estimation (MLE) is a standard approach to training a neural text generation model, e.g. Transformer (Vaswani et al., 2017), to generate human-like text. However, existing generation systems often suffer from the low-diversity problem (Holtzman et al., 2020; Welleck et al., 2020), which leads to generating dull and repetitive text. This problem unavoidably affects the overall quality of the generations.

We conclude that the low-diversity problem is mainly manifested in two aspects: form and content (Fu et al., 2020; Holtzman et al., 2020; Tevet and Berant, 2021). As shown Table 1, the low form diversity can be reflected in repeating some words, using similar lexicon and syntax, and more. The low content diversity can be expressed as a single and dull content with nothing different.

Several feasible fixes have been proposed, such as post-hoc sampling strategies including temperature (Caccia et al., 2020), top-\(k\) (Fan et al., 2018), and nucleus sampling (Holtzman et al., 2020). Recently, some works suggest that it is the maximizing likelihood itself that should account for the low-diversity problem (Holtzman et al., 2020; Welleck et al., 2020). Holtzman et al. (2020) think that MLE can not adequately capture the rich diversity and expression in human language. Choi et al. (2020) argue that the imbalanced token distribution inherent in natural language even worsens the low-diversity problem. Based on these analysis, many training-based methods have been proposed. Welleck et al. (2020) propose the unlikelihood training to penalize repetition with auxiliary losses. Jiang et al. (2019) propose to utilize dynamically scaling losses conditioned on the token frequency in the training phase. Choi et al. (2020) factorize the probability distribution and design an elaborate token cluster algorithm for a balanced training.

Though those encouraging progress has been made, we argue that current training-based methods only take plain contextual features to promote diversity, rarely considering incorporating syntactic structure clues. For example, when humans are writing articles, it is natural to predetermine the part-of-speech (POS) before giving the next token. Existing studies have verified that incorporating POS can improve the translation quality in neural machine translation (NMT) (Sennrich and Haddow, 2016a; Yang et al., 2021). Intuitively, since the vocabularies of different POS vary a lot, the diversity of POS will certainly lead to the diversity of text. Unfortunately, we observe that existing methods with no consideration of the inner POS structure fail to learn the diversity of POS in human language (shown in Table 3).

All these factors motivate us to address the low-diversity problem with the guidance of POS. Thus, in this work, we first present the POS Guided Softmax (Figure 1), building upon a hybrid decoder that predicts two posterior probabilities: (i) next-POS, and (ii) next-token from the vocabulary of
the target POS. Our work shows that, following the
POS clue, our model can gain a deeper insight into
text’s syntactic structure. Thereafter, we propose
a POS Guided Sampling to improve the diversity
of generated text lexically and syntactically while
maintaining comparable quality.

To sum up, the contributions of our work are
three-fold. (i) We introduce a novel POS Guided
Softmax, incorporating POS tags as the observed
discrete decisions to improve text generation. (ii)
Based on POS Guided Softmax, POS Guided Sam-
ping is proposed to promote text diversity effec-
tively without degrading quality. (iii) We conduct
extensive experiments on language modeling and
paraphrase generation. Experimental results and
human evaluation show that our model can easily
adapt to different downstream tasks and generate
text with high diversity as well as quality.

2 Related Works

2.1 Diversity-promoting Methods

In order to tackle the low-diversity problem, prior
studies either introduce a decoding-based method
or a training-based method.

Decoding-based Methods. Although greedy
search and beam search are well known decod-
ing strategies for neural text generation, Holtzman
et al. (2020) have shown that these methods always
generate generic, repetitive, and awkward words.
Kulikov et al. (2018) and Vijayakumar et al. (2018)
have proposed several variants of beam search as
alternatives. Recently, stochastic decoding meth-
ods have been widely used, and some studies pro-
ses to sample from a truncated and renormalized
Softmax distribution. Top-k sampling (Fan et al.,
2018) only samples from the top-k most probable
tokens. Nucleus sampling (Holtzman et al., 2020)
only samples from the smallest set whose cumu-
lativity probability is at least $\alpha$. However, those
decoding-based methods are lack of controllabil-
ity. Combined with above methods, our POSG
can further promote diversity using POS as a more
controllable clue.

Training-based Methods. As a standard approach
to training a neural text generation model, MLE has
been proved to be defective. Choi et al. (2020) have
shown that MLE may mislead the model because
of the imbalanced token distribution. Thus, they
design a greedy approach MefMax and factorize
Softmax to ensure a balanced training according
to the word frequency. FACE (Jiang et al., 2019)
utilizes the target word frequency to modify the
cross-entropy loss with a frequency-based weight
factor. Welleck et al. (2020) introduce an unlikeli-
hood loss to implicitly reduce the frequent tokens
and potential repeats. Other approaches, such as
negative training (He and Glass, 2020), reinforce-
ment learning (Shirai et al., 2020), and imitation
learning (Zhou and Lampouras, 2020), have re-
cently been applied to promote the diversity during
the training phase. All above training-based meth-
ods only learn from plain contextual features, while
ignoring other linguistic features. Our focus is on
leveraging POS features to guide both phases of
training and decoding.

2.2 POS in Text Generation

Previous works, which leverage POS for text gen-
eration, can be summarized as follows:

POS in Encoding. A branch of previous works
(He et al., 2019; Sennrich and Haddow, 2016b;
Wray et al., 2019) explore to adopt POS on the encoding side to help language understanding and generation. Sennrich and Haddow (2016b) concatenate the embeddings of POS tags with sentence features to improve the translation quality. For the image caption generation, He et al. (2019) use POS tags to control the fusion of the image features and the related word embeddings. Wray et al. (2019) enrich the encoding with POS of the accompanying captions for cross-modal search tasks.

**POS in Decoding.** The second line of studies directly model the POS structure during decoding. Su et al. (2018) introduce a hierarchical decoder that relies on teacher forcing to learn different POS patterns on different layers. Deshpande et al. (2019) use POS tag sequences as summaries to implicitly drive image caption generation. Yang et al. (2019) treat POS tags as latent variables in NMT and optimize the model by Expectation Maximization (EM). Yang et al. (2021) employ POS sequences to constrain the non-autoregressive generation (NAG) modes to alleviate the multi-modality problem. However, all the previous studies only focus on a single specific task and leverage POS as hidden decoding features (Deshpande et al., 2019; Yang et al., 2019), teacher forcing techniques (Su et al., 2018; Bugliarello and Elliott, 2021) or NAG plannings (Yang et al., 2021) in order to improve the generic quality of generated texts, while our proposed methods regard POS tags as observed sequential variables and directly model the POS distribution during both phases of training and decoding with the goal of improving text diversity.

To our best knowledge, we are the first to introduce an explicit POS-guided generation method as a generic way to promote text diversity while maintaining quality.

### 3 Language Modeling

The goal of language models is to assign a probability to text (i.e., word sequence) $\mathbf{x} = [x_1, \ldots, x_T]$, where each $x_t$ in the sequence is a token from a vocabulary $\mathcal{V}$, i.e., $x_t \in \mathcal{V}$, and $T \in \mathbb{N}$. We train the language models to learn a distribution $p_\theta (\mathbf{x})$ with the goal to fit the ground-truth distribution $p_* (\mathbf{x})$ for all $\mathbf{x}$. Specifically, when the language model is a neural network, $\theta$ is regarded as the model parameters of the neural network, and we can factorize $p_\theta (\mathbf{x})$ as $p_\theta (\mathbf{x}) = \prod_{t=1}^T p_\theta (x_t | \mathbf{x}_{<t})$. The conventional approach for learning the language model parameters $\theta$ is to maximize the log-likelihood by minimizing:

$$ L_{MLE} (\theta) = -\sum_{t=1}^T \log p_\theta (x_t | \mathbf{x}_{<t}) ,$$

$$ p_\theta (x_t | \mathbf{x}_{<t}) = \frac{\exp h_t \top \mathbf{w}_{x_t}}{\sum_{x \in \mathcal{V}} \exp h_t \top \mathbf{w}_x} ,$$

where $h_{t-1}$ is a hidden state of the context $\mathbf{x}_{<t}$, and $\mathbf{w}_{x_t}$ is the output embedding vector for $x_t \in \mathcal{V}$.

### 4 Methodology

POS-G is designed to exploit syntactic structure, i.e., POS tags for text generation in both the training and decoding phases. Specifically, giving text sequence $\mathbf{x} = [x_1, \ldots, x_T]$, we first use off-the-shelf POS tagger (Manning et al., 2014) to annotate corresponding POS sequence $\mathbf{\rho} = [\rho_1, \ldots, \rho_T]$, where each $\rho_t$ is a POS tag from the POS vocabulary $\mathcal{P}$, i.e., $\rho_t \in \mathcal{P}$, and $T \in \mathbb{N}$. We define all the tokens whose POS is $\rho$ as a vocabulary $\mathcal{V}_\rho$, where $\mathcal{V}_\rho \subset \mathcal{V}$.

#### 4.1 POS Guided Softmax

Figure 1 illustrates the core idea of our POS Guided Softmax. Given a context, there exists various choices for the next POS, which can be modeled as the next POS distribution. For the context “no one knows”, the next possible POS includes WH-pronoun (WP), preposition (IN), etc. For example, if WP is predicted as the next POS, the model will decode the next token from the WP vocabulary ($\mathcal{V}_{WP}$) with the token distribution of WP. Consequently, the complete sequence can be “no one knows what will happen”. For another case, if IN is predicted as the next POS, the next token will be decoded from $\mathcal{V}_{IN}$ with the corresponding token distribution. Then, the sequence may end up saying “no one knows until it finally happens”. This example also shows that the different choices of POS at each time step can result in vastly different generated text, thus promoting text diversity.

Following the core idea, we assume that the decoding process can be divided into two stages: for each time $t$, a POS tag $\rho_t$ is predicted first, and then the model decodes next-token $x_t$ from $\mathcal{V}_{\rho_t}$. Therefore, the joint conditional probability of $x_t$ and its corresponding POS tag $\rho_t$ is formulated as:

$$ p_\theta (x_t, \rho_t | \mathbf{x}_{<t}) = p_{\rho_1} (\rho_t | \mathbf{x}_{<t}) \times p_{\rho_2} (x_t | \rho_t, \mathbf{x}_{<t}) ,$$

where $p_{\rho_1} (\rho_t | \mathbf{x}_{<t})$ is the next-POS probability and $p_{\rho_2} (x_t | \rho_t, \mathbf{x}_{<t})$ is the next-token probability conditioned on $\rho_t$. These probabilities are defined

$\text{minimizing:}$

$$ L_{MLIE} (\theta) = -\sum_{t=1}^T \log p_{\theta} (x_t | \mathbf{x}_{<t}) ,$$

$$ p_{\theta} (x_t | \mathbf{x}_{<t}) = \frac{\exp h_t \top \mathbf{w}_{x_t}}{\sum_{x \in \mathcal{V}} \exp h_t \top \mathbf{w}_x} ,$$

where $h_{t-1}$ is a hidden state of the context $\mathbf{x}_{<t}$, and $\mathbf{w}_{x_t}$ is the output embedding vector for $x_t \in \mathcal{V}$. 

$\text{minimizing:}$
We propose POS Guided Sampling based on POS Guided Softmax. Consistent with POS Guided Softmax, the key idea is to divide the whole sampling process into two stages: POSt sampling and token sampling. In POS sampling, we first sample a POS, and then in token sampling, we use the sampled POS to control the sampling of tokens. Note that arbitrary sampling strategies can be adopted to both the POS sampling and token sampling. Here, we take top-k sampling for POS sampling, and nucleus sampling for token sampling as an example, and then we can formulate our POS Guided Sampling as follows:

\[
p_k'(x_t | x_{<t}) = \sum_{\rho_t \in \mathcal{P}} [p_{\theta_1}' (\rho_t | x_{<t}) \times p_{\theta_2}' (x_t | \rho_t, x_{<t})],
\]

\[
p_{\theta_1}' (\rho_t | x_{<t}) = \begin{cases} 
\sum_{x_t \in \mathcal{V}_{\rho_t}} p_\rho (x_t | \rho_t, x_{<t}), & \text{if } x_t \in \mathcal{V}_{\rho_t}, \\
0, & \text{otherwise}.
\end{cases}
\]

\[
p_{\theta_2}' (x_t | \rho_t, x_{<t}) = \begin{cases} 
\sum_{x_t \in \mathcal{V}_{\rho_t}} p_\rho (x_t | \rho_t, x_{<t}), & \text{if } x_t \in \mathcal{V}_{\rho_t}, \\
0, & \text{otherwise}.
\end{cases}
\]

\[
\alpha_{\text{token}}(\mathcal{P}) \text{ and } \alpha_{\text{token}}(0 < \alpha_{\text{token}} < 1) \text{ are the hyperparameters for the sampling of POS and token, respectively. For other sampling strategies used in POS sampling and token sampling, POS Guided Sampling can be similarly defined.}

4.2 POS Guided Sampling

As mentioned before, we think of POS tags as observed sequential variables, which also contributes to the model interpretability and controllability. Then, the final next-token distribution can be formulated as: \( p_\theta (x_t | x_{<t}) = \sum_{\rho_t \in \mathcal{P}} p_\theta (x_t | \rho_t, x_{<t}) \). Note that some tokens may have more than one POS, and \( p_\theta (x_t, \rho_t | x_{<t}) = 0 \) for \( x_t \notin \mathcal{V}_{\rho_t} \). Since the number of POS in a specific language family is fixed, there is no problem of insufficient exploration in variables’ space.

As mentioned before, we think of POS tags as observed sequential variables and extend the training text set with annotated POS sequences, so we define the POS guided training objective as follows:

\[
\mathcal{L}_{\text{POS-Guided}} (\theta) = - \sum_{t=1}^{T} [\log p_{\theta_1} (\rho_t | x_{<t}) + \log p_{\theta_2} (x_t | \rho_t, x_{<t})].
\]
5 Experiments

We systematically evaluate our proposed methods on language modeling task (Section 5.2) and paraphrase generation task (Section 5.3).

5.1 Experimental Setup

Model Architecture Since our proposed methods are architecture agnostic, we implement POS Guided Softmax on the Transformer (Vaswani et al., 2017), a widely used architecture for neural text generation. Details of the experimental setup for each task are shown in Appendix A.

Baseline Models We compare our POS Guided Softmax and Sampling (POSG) with the following baselines: (i) Maximum likelihood estimation (MLE), a standard approach for neural text generation. (ii) Frequency-Aware Cross-Entropy (FACE) (Jiang et al., 2019) dynamically weights the cross-entropy losses conditioned on the token frequency. (iii) Frequency Factorization Softmax (F²-Softmax) (Jiang et al., 2019) factorizes the standard Softmax based on the token frequency. (iv) Unlikelihood training (UL) (Welleck et al., 2020) is to enhance the log-likelihood loss with an unlikelihood loss that penalizes the generation of repeated tokens. (v) We further implement two task-specific baselines: Mixture of Softmaxes (MoS) (Yang et al., 2018) for language modeling, Syntax Guided Controlled Paraphraser (SGCP) (Kumar et al., 2020) for paraphrase generation. Note that decoding-based methods, including top-k and nucleus sampling, can be directly compared to POSG, when they are applied to MLE. The details will be described in the sections of Generation Details.

5.2 Language Modeling

Dataset We performed experiments on the Wikitext-103 dataset (Merity et al., 2017) for language modeling. In order to train our POS Guided Softmax, we need the corresponding POS tags. We use the Stanford CoreNLP’s POS tagger (Manning et al., 2014) to annotate words in Wikitext-103 with XPOS tags (Hornby et al., 2017). In our implementation, there are 45 different POS tags in total.

Generation Details We conduct the text completion task to evaluate models on the test set. Specifically, for each sample, we truncate 50 tokens as the prefix, and then guide model to decode following 100 tokens as the continuation from the given prefix. Finally, there are 1536 prefixes in the test set. We use stochastic decoding to generate text. Note that all the baselines have only one sampling stage, i.e., token sampling, while our POSG has an additional POS sampling. To reach a good trade-off between quality and diversity, we adopt nucleus sampling with α^{(token)} = 0.5 for token sampling (for all models including our POSG and baselines). For our POSG, we adopt top-k sampling in POS sampling, since the size of the POS vocabulary P is much smaller than the total token vocabulary. We then conduct a grid search to find the k^{(POS)} whose generated continuations have the smallest reverse language model score (Semeniuta et al., 2018) on the validation set. k^{(POS)} is finally set to 20. Some generated cases are shown in Appendix D.

Metrics Following Choi et al. (2020), we evaluate the generated text with two sets of metrics: (i) Diversity: We use Self-BLEU (Zhu et al., 2018) which is calculated by computing BLEU (Papineni et al., 2002) of each generated text with all other generations as references. We also compute the generated continuations’ unique tokens (Uniq), distinct n-gram (Distinct-n). We also use repetition (Rep) (Holtzman et al., 2020), the percentage of continuations ending with a repetition loop, to evaluate text diversity. (ii) Quality: We measure the perplexity (PPL) (Mnih and Teh, 2012), KL-Divergence (KLD) (Kullback, 1997) on unigram distributions, and MS-Jaccard (Montañahet al., 2019) on n-gram. All of above metrics are calculated between the generated continuations as hypotheses and the ground truths as references.

Automatic evaluation Table 2 shows the automatic evaluation results comparing different models on the language modeling task. In terms of Self-BLEU4, Rep, and Distinct-n, our POSG performs much better than all the baselines, indicating that our proposed model can generate diverse text effectively. The FACE also performs well, and it achieves the best in Uniq. However, by checking the outputs (Table 13 in Appendix D), we find that FACE produces more incoherent text that is hard to understand.

Since training-based methods including ours make a trade-off between the text diversity and the likelihood of ground truth, MLE gets the lowest PPL. However, the optimal or second best results of quality metrics confirm that POSG can still main-
We further conduct a correlation test to verify that the text diversity is closely correlated with the POS diversity. We first annotate all the generated text with the POS tagger, and define a \( n \)-POS to be contiguous \( n \) POS tags from the annotated POS tag sequence. Then, we can describe the degree of POS diversity by calculating the proportion of the distinct \( n \)-POS. Table 3 presents results of distinct \( n \)-gram and \( n \)-POS with corresponding Pearson product-moment correlation coefficient (PPMCC). \( n \)-P and \( n \)-G where \( n \in \{1, 2, 3\} \) are abbreviated notations for \( n \)-POS and \( n \)-gram.

### 5.3 Paraphrase Generation

**Dataset**: We use the the ParaNMT-50M\(^3\) dataset (Wieting and Gimpel, 2018) for paraphrase generation. For better training, we follow Goyal and Durrett (2020) to filter this dataset, and finially get 1.6 million paraphrase pairs with both high quality and diversity. We also use Stanford CoreNLP to tokenize the text and get corresponding POS tags.

**Generation Details**: We conduct the standard sequence-to-sequence paraphrase generation for testing. Note that, during inference, SGCP needs a corresponding exemplar sentence to paraphrase the input sentence, while our model does not. So, for a fair comparison, we prune the exemplar tree to the height \( \max(3, H_{\text{max}} - 4) \) to reduce the impact from exemplar sentence, where \( H_{\text{max}} \) is the height of the full constituency tree of the exemplar sentence. We use the test set provided in the work of SGCP\(^4\) that contains 800 paraphrase pairs and correspond exemplar sentences for inference. For fair comparison, we closely follow Kumar et al. (2020) for paraphrase generation.

---

\(^3\)https://drive.google.com/file/d/1rbF3daJjCsal-fu2GAnEj3d2FBXo1ugD/view

\(^4\)https://github.com/malllabiisc/SGCP
### Table 5: Automatic evaluation results for different models on the paraphrase generation task. Numbers \( n \in \{1, 2, 3\} \) in the column heads under Distinct refer to \( n \)-gram. (Bold: the best; Underline: the second best).

<table>
<thead>
<tr>
<th>Models</th>
<th>Self-WER↑</th>
<th>Self-BLEU4↓</th>
<th>Distinct↑</th>
<th>BERTScore↑</th>
<th>BLEU↑</th>
<th>ROUGE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=1</td>
<td>n=2</td>
<td>n=3</td>
<td>1</td>
<td>2</td>
<td>L</td>
</tr>
<tr>
<td>MLE</td>
<td>74.2</td>
<td>25.1</td>
<td>78.4</td>
<td>82.8</td>
<td>78.5</td>
<td>47.4</td>
</tr>
<tr>
<td>FACE</td>
<td>73.0</td>
<td>25.0</td>
<td>78.9</td>
<td>83.6</td>
<td>79.6</td>
<td>48.1</td>
</tr>
<tr>
<td>F2-Softmax</td>
<td>76.4</td>
<td>28.0</td>
<td>78.2</td>
<td>83.0</td>
<td>79.5</td>
<td><strong>53.9</strong></td>
</tr>
<tr>
<td>UL</td>
<td>77.2</td>
<td>21.2</td>
<td>80.1</td>
<td><strong>85.3</strong></td>
<td>80.9</td>
<td>36.0</td>
</tr>
<tr>
<td>SGCP</td>
<td>83.0</td>
<td>28.6</td>
<td>81.9</td>
<td>82.6</td>
<td>77.7</td>
<td>47.9</td>
</tr>
<tr>
<td>POSG</td>
<td><strong>89.7</strong></td>
<td><strong>19.6</strong></td>
<td><strong>82.1</strong></td>
<td><strong>85.3</strong></td>
<td><strong>81.8</strong></td>
<td>48.3</td>
</tr>
</tbody>
</table>

### Table 6: Human evaluation on paraphrase generation. * denotes statistical significance compared with POSG (Mann-Whitney u-test, \( p < 0.1 \)).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>2.92</td>
<td>3.34*</td>
<td>3.09*</td>
<td></td>
</tr>
<tr>
<td>FACE</td>
<td>2.91</td>
<td>2.58*</td>
<td>3.60</td>
<td>3.35</td>
</tr>
<tr>
<td>F2-Softmax</td>
<td>2.77*</td>
<td>2.57*</td>
<td>3.59</td>
<td><strong>3.38</strong></td>
</tr>
<tr>
<td>UL</td>
<td>3.00</td>
<td>2.68*</td>
<td>3.37*</td>
<td>3.17*</td>
</tr>
<tr>
<td>SGCP</td>
<td>2.74*</td>
<td>2.67*</td>
<td>3.50</td>
<td>3.21*</td>
</tr>
<tr>
<td>POSG</td>
<td><strong>3.02</strong></td>
<td><strong>2.79</strong></td>
<td>3.58</td>
<td>3.35</td>
</tr>
</tbody>
</table>

### Figure 2: Quality-diversity trade-off for different models on paraphrase generation. The x-axis measures BLEU4 for quality, and the y-axis measures negative Self-BLEU4 for diversity. Both are the bigger the better.

To generate paraphrase using beam search for all the models with beam size 10. For the sampling hyperparameter in POS sampling, we also conduct a grid search, and \( k^{(\text{POS})} \) is finally set to 5. Some generated cases are shown in Appendix D.

### Metrics
We also evaluate the generated paraphrases with two sets of metrics, (i) **Diversity**: To assess how different the generated paraphrases are compared to the original sentences, we calculate BLEU and Word Error Rate (WER) (Goyal and Durrett, 2020) between generated paraphrases and input sentences. We denote them as Self-BLEU and Self-WER (see Appendix A.3 for the difference with the Self-BLEU in language modeling) and Self-WER, respectively. We also compute the generated paraphrases’ distinct \( n \)-gram (Distinct-\( n \)) to evaluate text diversity. (ii) **Quality**: we calculate BLEU score on \( n \)-gram to evaluate the closeness of the generated paraphrases to references. Besides, we use the BERTScore (Zhang et al., 2020) to measure the semantic consistency between generated paraphrases and input sentences. We also compute ROUGE-1,2,L between the generated and the reference to evaluate the generation quality.

### Automatic evaluation
The experimental results on the paraphrase generation task are shown in Table 5. Our proposed model outperforms other base-lines on all the diversity metrics. In terms of quality metrics, our POSG performs better than MLE, FACE, and UL, while the best model in quality, i.e., F2-Softmax performs badly in diversity. Moreover, compared with other syntax-guided models, i.e., SGCP, our model performs much better in diversity and has a comparable performance in quality. This further confirms that our model can effectively promote text diversity without the help of exemplars.

To make a more intuitive comparison, we further apply stochastic decoding for different models, and tune the sampling hyperparameters to generate different sets of paraphrases. Then, we calculate BLEU4 and Self-BLEU4 scores for these sets, and draw the quality-diversity trade-off in Figure 2. Clearly, POSG surpasses all the baselines with a significant gap. These results confirm that our model can produce equally high-quality text that is more diverse, and vice versa.

### Human Evaluation
We also conduct a human evaluation for the generated paraphrases. 100 examples are randomly sampled from each models’ outputs, respectively. Each of them are evaluated by five workers from the following four aspects:
Table 7: Results of ablation study on the language modeling task. Note that PPL measures the ability of the model to generate fluent text, which is not affected by the sampling strategy.

<table>
<thead>
<tr>
<th>Models</th>
<th>Self-BLEU4 ↓</th>
<th>Rep ↓</th>
<th>Uniq ↑</th>
<th>Distinct ↑</th>
<th>PPL ↓</th>
<th>KLD ↓</th>
<th>MS-Jaccard ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=1</td>
<td>n=2</td>
<td>n=3</td>
<td>n=1</td>
<td>n=2</td>
<td>n=3</td>
<td></td>
</tr>
<tr>
<td>POSG</td>
<td>34.1</td>
<td>0.000</td>
<td>13.8k</td>
<td>60.2  88.8</td>
<td>94.3</td>
<td>34.4</td>
<td>1.17  62.2  40.7  25.9</td>
</tr>
<tr>
<td>w/o POSG-Sampling</td>
<td>40.6</td>
<td>0.841</td>
<td>13.1k</td>
<td>55.2  83.0</td>
<td>90.6</td>
<td>34.4</td>
<td>1.29  56.9  37.5  24.5</td>
</tr>
<tr>
<td>MLE</td>
<td>46.9</td>
<td>1.86</td>
<td>11.7k</td>
<td>50.2  77.2</td>
<td>86.2</td>
<td>32.7</td>
<td>1.34  56.9  38.2  25.4</td>
</tr>
</tbody>
</table>

Table 8: Results of ablation study on the paraphrase generation task.

<table>
<thead>
<tr>
<th>Models</th>
<th>Self-WER↑</th>
<th>Self-BLEU↓</th>
<th>Distinct↑</th>
<th>BERTScore↑</th>
<th>BLEU↑</th>
<th>ROUGE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=1</td>
<td>n=2</td>
<td>n=3</td>
<td>1</td>
<td>2</td>
<td>L</td>
</tr>
<tr>
<td>POSG</td>
<td>89.7</td>
<td>19.6</td>
<td>82.1  85.3</td>
<td>81.8</td>
<td>48.3</td>
<td>9.79  40.3  17.1  39.4</td>
</tr>
<tr>
<td>w/o POSG-Sampling</td>
<td>87.6</td>
<td>24.1</td>
<td>78.0  80.5</td>
<td>78.5</td>
<td>52.6</td>
<td>11.1  40.9  19.7  42.4</td>
</tr>
<tr>
<td>MLE</td>
<td>74.2</td>
<td>25.1</td>
<td>78.4  82.8</td>
<td>78.5</td>
<td>47.4</td>
<td>9.81  38.1  16.9  38.8</td>
</tr>
</tbody>
</table>

Table 9: Results of controllability analysis on the paraphrase generation task. “×n” means that we manually multiply the probability of “Adjective” by n.

<table>
<thead>
<tr>
<th>Adjective Probability</th>
<th>Adj. per Sentence</th>
<th>Self-BLEU↓</th>
<th>BLEU↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>×0.1</td>
<td>0.43</td>
<td>20.2</td>
<td>9.47</td>
</tr>
<tr>
<td>×1</td>
<td>0.66</td>
<td>19.6</td>
<td>9.79</td>
</tr>
<tr>
<td>×10</td>
<td>1.04</td>
<td>18.7</td>
<td>9.45</td>
</tr>
</tbody>
</table>

6 Analysis

6.1 Interpretability

Compared with one-stage sampling such as top-k sampling, POSG will lead to the entropy increasing of a language model’s distribution, and thus lead to more diverse outputs (see Appendix B for the proof, Appendix C.1 for experimental results).

6.2 Controllability

Our proposed POSG first samples a POS, and then samples a token from the vocabulary of the previously predicted POS. Therefore, we can control the POS sampling stage by forcing the probability of some specific POS to be higher or lower. For example, on the paraphrase generation task, we can multiply the probability of “Adjective” (“JJ”) and renormalize by dividing by the sum, aiming at generating more descriptive style paraphrases.

The results are shown in Table 9. These results confirm that by leveraging POS as an observed and controllable clue, the generated text can be successfully modulated with negligible effect on quality and diversity (see Appendix C.2 for cases).

7 Conclusion

In this paper, we have introduced POS Guided Softmax and Sampling, simple but effective methods to address the low-diversity problem in text generation. POSG guides models to capture contextual and syntactical information by leveraging POS as an observed and controllable clue in both the training and decoding phases. Experimental results and human evaluation on language modeling and paraphrase generation have demonstrated the effectiveness of our methods.
References


A Experimental Setup

A.1 Dataset

The dataset statistics of Wikitext-103 and ParaNMT-50M are reported in Table 10 and Table 11, respectively.

Since ParaNMT-50M is generated by back-translation, the dataset has provided translation scores to measure the quality of back-translation, that a low translation score means semantically inconsistent, while a high translation score usually accompanies low diversity. Therefore, we only keep the paraphrase pairs whose translation scores are between 0.7 and 0.8. Moreover, for better training, we remove the sentences that are less than 10 tokens. Finally, we get a filtered dataset containing 1.6 million paraphrase pairs with both high quality and diversity.

For language modeling, we use the original settings of Wikitext-103 dataset for training, validation, and test set splitting. For paraphrase generation, we use the filtered training, validation set of ParaNMT-50M, and the test set provided in the work of SGCP. It is worth to mention that Wikitext-103 is under the CC BY-SA 3.0 license, and ParaNMT-50M is under the CC-BY license.

<table>
<thead>
<tr>
<th>#Articles</th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>28,475</td>
<td>60</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>#Tokens</td>
<td>103,227,021</td>
<td>217,646</td>
<td>245,569</td>
</tr>
</tbody>
</table>

Table 10: Statistics of Wikitext-103.

<table>
<thead>
<tr>
<th>#Sentence</th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,640,709</td>
<td>3,000</td>
<td>800</td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Statistics of ParaNMT-50M.

A.2 Architectures and Hyperparamters

For the language modeling task, we use a 12-layer Transformer Decoder with 8 attention heads, embedding dimension 512, and projection dimension 2048. For the paraphrase generation task, we use a 6-layer Transformer Encoder and Decoder with the same other settings. All the algorithms are implemented in Pytorch and trained on a machine with 8 NVIDIA GTX 2080Ti GPUs for 10 epochs with the hyperparameters reported in Table 12.

We choose the architecture settings and batch sizes according to the GPU memory constraint.

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Wikitext-103</th>
<th>ParaNMT-50M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary size</td>
<td>267,735</td>
<td>100,000</td>
</tr>
<tr>
<td>Batch size</td>
<td>12</td>
<td>96</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>Fine-tuning LR</td>
<td>0.00001</td>
<td>0.00001</td>
</tr>
<tr>
<td>Fine-tuning step</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>Gradient clipping</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Dropout</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
<td>Adam</td>
</tr>
<tr>
<td>-β₁</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>-β₂</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>-ε</td>
<td>1e-8</td>
<td>1e-8</td>
</tr>
</tbody>
</table>

Table 12: Hyperparameter settings for different datasets.

Note that we use FACE-OPR among the four variants of FACE, and we train it in the way of fine-tuning with corresponding finetuning LR and finetuning step. Additionally, we use 7 mixture components in MoS.

A.3 Metrics

Note that, the calculations of Self-BLEU are different for language modeling and paraphrase generation. This is because the typical definitions of Self-BLEU for these two different task are indeed different. For language modeling, Self-BLEU (Zhu et al., 2018) is a metric to evaluate the inner diversity of the generated data, while for paraphrase generation, Self-BLEU (Cao and Wan, 2020) is used to evaluate the degree to which the generated paraphrases are different from the original sentence.

B Proof

We prove that our POS Guided Softmax and Sampling can certainly generate more diverse text than the one-stage sampling, top-k sampling as an example.

In information theory, the entropy of a random variable is the average level of “information”, “surprise” in the variable’s possible outcomes. Therefore, we can use the entropy of a language model’s distribution $p(x)$ to measure its diversity. We denote the entropy of $p(x)$ as $H(p)$:

$$H(p) = - \sum_{x \in V} p(x) \log p(x).$$

The increase of $H(p)$ means the increase of diversity.

For example, compared with greedy search, diversity-promoting sampling methods, such as top-k sampling can increases $H(p)$ from

$$-p(x_{\text{max}}) \log p(x_{\text{max}})$$

to

$$- \sum_{x \in V_k} \frac{p(x)}{|V_k|} \log \frac{p(x)}{|V_k|},$$

where $x_{\text{max}}$ is the token with the max probability, $V_k$ is the set of top-k most probable tokens, $Z_k = \sum_{x \in V_k} p(x)$, and obviously $x_{\text{max}} \in V_k$. 

Note that we use FACE-OPR among the four variants of FACE, and we train it in the way of fine-tuning with corresponding finetuning LR and finetuning step. Additionally, we use 7 mixture components in MoS.
According to the Log sum inequality, it follows:

\[
H(p)^{(\text{POS})} = - \sum_{x \in V} \sum_{\rho \in \mathcal{P}} p'(x, \rho) \log \sum_{\rho \in \mathcal{P}} p'(x, \rho) = - \sum_{x \in V} \sum_{\rho \in \mathcal{P}} p'(x | \rho) \log p'(x | \rho) \log p'(x | \rho)
\]  

(9)

where \( p'(x, \rho) \) is defined in Equation 2, \( p'(x | \rho) \) and \( p'(x | \rho) \) are defined in Equation 5. Again, according to the Log sum inequality, it follows:

\[
H(p)^{(\text{POS})} \geq - \log |\mathcal{P}|
\]  

(10)

For the sake of briefness and fairness, we assume that our POSG adopts pure sampling in the first sampling stage (POS Sampling), and adopts top-k sampling with the same \( k \) in the second sampling stage (Token Sampling). So, \( p'(\rho) = p(\rho) \) for \( \rho \in \mathcal{P} \), while

\[
p'(x | \rho) = \begin{cases} 
\frac{p(x, \rho)}{Z_k}, & \text{if } x \in V_{k,\rho} \\
0, & \text{otherwise}
\end{cases}, \quad Z_k = \sum_{x \in V_{k,\rho}} p(x)
\]

(8)

Note that, in our paper, we denote all the tokens whose POS is \( \rho \) as a vocabulary \( V_\rho \), and here, \( V_{k,\rho} \subseteq V_\rho \).
<table>
<thead>
<tr>
<th>Models</th>
<th>Self-WER↑</th>
<th>Self-BLEU4↓</th>
<th>Distinct↑</th>
<th>BERTScore↑</th>
<th>BLEU4↑</th>
<th>ROUGE↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=1</td>
<td>n=2</td>
<td>n=3</td>
<td>1</td>
<td>2</td>
<td>L</td>
</tr>
<tr>
<td>top-k</td>
<td>100.8</td>
<td>102.1</td>
<td>+1.3</td>
<td>39.4</td>
<td>43.3</td>
<td>+3.9</td>
</tr>
<tr>
<td>POSG</td>
<td>13.6</td>
<td>13.6</td>
<td>+0.0</td>
<td>6.49</td>
<td>7.71</td>
<td>+1.22</td>
</tr>
<tr>
<td></td>
<td>86.9</td>
<td>86.9</td>
<td>+0.0</td>
<td>88.9</td>
<td>88.2</td>
<td>+2.9</td>
</tr>
<tr>
<td></td>
<td>83.7</td>
<td>83.4</td>
<td>-0.7</td>
<td>33.5</td>
<td>36.4</td>
<td>+2.0</td>
</tr>
<tr>
<td></td>
<td>12.1</td>
<td>14.1</td>
<td>-0.3</td>
<td>12.1</td>
<td>14.1</td>
<td>+2.6</td>
</tr>
<tr>
<td></td>
<td>32.3</td>
<td>34.9</td>
<td></td>
<td>32.3</td>
<td>34.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.5</td>
<td>36.4</td>
<td></td>
<td>33.5</td>
<td>36.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12.1</td>
<td>14.1</td>
<td></td>
<td>12.1</td>
<td>14.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.3</td>
<td>34.9</td>
<td></td>
<td>32.3</td>
<td>34.9</td>
<td></td>
</tr>
</tbody>
</table>

Table 14: Results of POSG and one-stage sampling (we use top-k sampling here) on the paraphrase generation task. Note that we tune the sampling hyper-parameters of both methods to reach the same level of diversity, and then compare the text quality.

**Source:** this is going to make good economic sense for the city.

**Reference:** that it would be good for the city in a certain economic sense.

**MLE:** this will be an economic sense for the entire city.

**FACE:** this will create a good economic point in the city.

**F^2-Softmax:** this will make sense of economic sense for the city.

**UL:** this will be considerable economic considerations for the city’s going to be economic point of the city.

**SGCT:** this will make economic sense for the city.

**POSG:** it is what makes good economic sense to the city.

is the set of top-k most probable tokens in $V_p$. Thus, $V_{p,k} \subseteq V_p$. Then, it follows:

$$H(p)^{(POS)} \geq -\log |P| - \sum_{x \in V} \sum_{\rho \in P} p(x | \rho) \log p(\rho)p'(x | \rho)$$

$$= -\log |P| - \sum_{\rho \in P} \sum_{x \in V_{p,k}} p(x | \rho) \frac{p(x | \rho)}{Z_2} \log p(\rho) \frac{p(x | \rho)}{Z_2}$$

$$= -\log |P| - \sum_{\rho \in P} \sum_{x \in V_{p,k}} \frac{p(x | \rho)}{Z_2} \log p(x | \rho)$$

(11)

Since $V_{k,p} \subseteq V_{p,k}$ and we use the same setting of $k$, i.e., $Z_2 \approx Z_k$, we can finally conclude from Equation 8 and Equation 11 that the lower bound of $H(p)^{(POS)}$ is greater than or equal to the lower bound of $H(p)^{(top-k)}$. When compared with other one-stage sampling strategies, this conclusion still holds, and can be proved in a similar way. Consequently, this will account for the effectiveness of our methods.

C.1 Compared with One-stage Sampling

We further conduct an analysis to test whether the traditional one-stage sampling can achieve the same level of diversity by increasing the randomness, e.g., using larger $k$ in top-$k$ sampling. On

the paraphrase generation task, we tune the sampling hyper-parameters in top-$k$ sampling and our POSG to reach the same level of diversity, and then compare the text quality. The results are shown in Table 14. In this experiment, POSG adopts top-$k$ sampling with $k^{(POS)} = 5$ in POS sampling, $k^{(token)} = 500$ in token sampling, while MLE adopts top-$k$ sampling with $k = 1000$. Obviously, our POSG significantly outperforms top-$k$ sampling on MLE in terms of quality metrics, while performing equally well in diversity. Therefore, we can conclude that, by increasing the randomness, the traditional one-stage sampling on MLE can finally achieve the same level of diversity as our POSG, but the quality of the generated text will seriously deteriorate. This further confirms the advantage of our methods over prior works.

C.2 Controllability Analysis Example

An example of the controllability analysis is provided in Table 16. When we control the probability of adjective increasing during the POS sampling stage, the generated paraphrase will contain correspondingly more adjectives.

D Case Study

Table 13 provides examples of text completion produced by our model and other baselines. It can be observed that MLE, F^2-Softmax, and MoS suf-
Table 17: Agreement analysis for annotators labels on the language modeling task.

<table>
<thead>
<tr>
<th>Models</th>
<th>Div.</th>
<th>Qua.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Krippendorff’s α</td>
<td>0.57</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Table 18: Agreement analysis for annotators labels on the paraphrase generation task.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Krippendorff’s α</td>
<td>0.54</td>
<td>0.37</td>
<td>0.71</td>
<td>0.63</td>
<td></td>
</tr>
</tbody>
</table>

Our work has developed generic generation methods to promote text diversity while maintaining comparable quality. Therefore, despite the contributions to better text generation, our proposed methods may be used to generate more human-like fake text. But the impacts are more apparent when considering deployed applications, while our proposed methods as the methodologies can not have any direct negative societal impacts. Moreover, all the datasets we used in our work are open source datasets. Wikitext-103 was extracted from Wikipedia, and ParaNMT-50M was created by the back-translation. Therefore, the data we used would not contain personally identifiable information or offensive content.

E Human Evaluation

We post the human evaluation questionnaire, as shown in Table 19 and Table 20, and then recruit five workers with sufficient high English skills. We pay each worker 60 US dollars, and let them complete the evaluation within a week. For both tasks, workers are given 100 randomly sampled inputs, and corresponding outputs from each model. Then, they need to score those outputs according to the description in the questionnaire. The term “diversity” in language modeling is typically regarded as a property of the collective outputs of a system, but it is really difficult for a human to remember such a large scale of outputs and give an overall score for a system. So we make a compromise that we asked the worker to rate the diversity of individual outputs, and intuitively the more diverse individual outputs are, the more diverse the system is.

We employ the Krippendorff’s alpha for the inter-annotator agreement analysis. As shown in Table 17 and Table 18, all the results are fair agreement (0.2 ≤ κ ≤ 0.4) or moderate agreement (0.4 ≤ κ ≤ 0.6).
The goal of this review is to evaluate the quality and diversity of generated texts. In this review, you will read an excerpt from Wikipedia with first 50 words as prefixes, and its corresponding 100-word continuations. You should rate the continuations between 1 - 5 in two ways:

(1) Diversity. The overall diversity of text can be evaluated from form (How to say it?) and content (What to say?). (1 = The continuation is always repeating some words, its sentences share the similar forms syntactically and lexically, and its content is dull; 5 = The continuation seldom repeats words, its sentences have various syntactical and lexical forms, and it contains different things related to the prefix)
(2) Quality. The overall quality of text can be evaluated in many different aspects, such as fluency, readability, coherence, and so on. (1 = The continuation is incoherent, difficult to understand, not related to the prefix, and has many syntactically and semantically errors; 5 = The continuation is coherent, easy to understand, related to the prefix, and grammatically correct)

You should score between 1 - 5, where 5 is best and 1 is worst. You can consider and make a final decision by comparing different continuations of the same prefix. These prefixes and continuations have been preprocessed by separating punctuation, and splitting conjunctions. And because of length constraints, they may be truncated in the middle of the text. Please ignore these when evaluating and do not allow them to affect your judgments.

For example, if you are given the following prefix and continuations:

**Prefix:** Infantry fought a series of hard battles. Simultaneously it had to send combat patrols to its rear to clear infiltrating North Koreans from Changnyong and from its supply road. On the morning of September 1 the 1st and 2nd Regiments of the NK 9th Division, in their...

<table>
<thead>
<tr>
<th>Continuation 1</th>
<th>Score: Diversity 3, Quality 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>second action around 380 yards &lt;unk&gt; 290 m &lt;unk&gt; from .395 , were assigned to the 7th Division. Meanwhile , the 2nd Regiment of the 27th Infantry , which had been sent out of Murmansk in late October , had moved to iconoclasts Pass . It was the first major US unit to advance across the mountains to the west , but it was not successful . On the morning of September 2 , the Eighth Army’s 2nd Battalion , 5th Cavalry Regiment was moved into a position about 10 miles &lt;unk&gt; 16 km &lt;unk&gt; from Highton .</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuation 2</th>
<th>Score: Diversity 1, Quality 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserve , were positioned to engage the North Korean troops . The 3rd Regiment of the NK 10th Division was to attack the North Korean forces . The 1st Regiment of the NK 12th Division was to attack the North Koreans from the north and east of the 2nd Division , and the 1st Regiment of the NK 15th Division was to attack the North Koreans from the south . The 1st Regiment of the NK 12th Division was to attack the North Koreans from the south . The 1st Regiment of the NK 15th Division was to attack the</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuation 3</th>
<th>Score: Diversity 1, Quality 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>position north of Wonju , were repeatedly pushed back by the ROK 3rd Division . At 08 : 00 am the units of the 1st Battalion attempted to attack . Kim of the 1st and 2nd Battalions attacked the 3rd and 3rd Battalions of the 2nd Battalion of the 3rd Battalion of the 3rd Battalions of the 1st Battalion of the 2nd Battalion of 2nd Battalion , 7th Marines on North , 7th Marines on Hill 60 . Task Force 51 and 9th Marines attacked Sangju ’s 1st Battalion of the 3rd Battalion of the 2nd Battalion , 1st Platoons</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Continuation 4</th>
<th>Score: Diversity 4, Quality 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>“ Series B ” Company , carried out three assaults on the Pusan on 29 September against three resistance groups that included the blacksmiths , truck commanders , and air support . They then conducted three raids into a line south of psalmody by the 2nd Battalion , 3rd Field Artillery Regiment . At the same time , units from the 3rd Infantry Division and the 3rd Marine Division advanced on all four sides of the road , while infantry units of the 2nd Infantry Division advanced on the northern slope . The 5th Marine Corps , in particular</td>
<td></td>
</tr>
</tbody>
</table>

**Analysis:** As for diversity, Continuation 1 gives various details about the "hard battles", and is of high diversity in the text form. But all the content of it is about the deployment of armies, which means low content diversity. Therefore, Continuation 1 gets 3 points in Diversity. Since there are some words difficult to understand (highlighted in red), Continuation 1 gets 3 points in Quality.

Continuation 2 keeps talking about only one single content, that is "some Regiment attacks the North Koreans from somewhere" (highlighted in orange). Although it is fluent, relevant, and gets high scores in Quality, Continuation 2 will receive the lowest score in Diversity due to its dull content.

Continuation 3 contains many useless repeating text (highlighted in blue), which makes the continuation incoherent and hard to understand, so it gets the lowest score in both Quality and Diversity.

Continuation 4 also states from many different aspects of the "hard battles", but compared to continuation 1, it is not that diverse (That’s why comparing different continuations can help to make a decision). Therefore, it gets 4 points in Diversity. In the meantime, high diversity of it also leads to some strange words in the text, and affects the overall quality. So, Continuation 4 can only get a mediocre score in Quality.

Table 19: Human evaluation questionnaire for language modeling.
The goal of this review is to evaluate the quality and diversity of text paraphrase dataset. In this review, you will be given an original sentence, and its corresponding paraphrases. You should rate the paraphrases between 1 - 5 in four ways:

1. Lexical Diversity: how lexically diverse are the generated sentences?
2. Syntactical Diversity: how syntactically diverse are the generated sentences?
3. Fluency: how fluent are the generated paraphrases?
4. Relevance: how semantically consistent are between generated paraphrases and the input sentences?

You should score between 1 - 5, where 5 is best and 1 is worst. You can consider and make a final decision by comparing different paraphrases of the same original sentence. These sentences have been preprocessed by converting all letters to lowercase, separating punctuation, and splitting conjunctions.

Please ignore this when evaluating and do not allow it to affect your judgments.

For example, if you are given the following original sentence and paraphrases:

**Original sentence**: by adopting rules that regulate the information about the foods and their nutritional value appearing on the label, the consumers will be able to make informed and meaningful choices.

**Paraphrase 1**: Score: Lexical Diversity 5, Syntactical Diversity 5, Fluency 3, Relevance 5

the rules will be able to adapt food and their nutritional values listed on the labelling of consumers will be able to be able to make informed and they are appropriate assessment.

**Paraphrase 2**: Score: Lexical Diversity 1, Syntactical Diversity 2, Fluency 1, Relevance 2

by adopting rules governing the information about food and relevance of foods and nutritional value of nutrition value that regulate the labelling, so that consumers.

**Paraphrase 3**: Score: Lexical Diversity 4, Syntactical Diversity 5, Fluency 1, Relevance 2

consumers can adopt rules to provide informed and nutrition value of the food and their nutritional values listed on the labelling, consumers will be able to enable consumers.

**Paraphrase 4**: Score: Lexical Diversity 1, Syntactical Diversity 1, Fluency 1, Relevance 1

by adopting rules that regulates the rule of food and their nutritional value of food and their nutritional value of their nutritional value to the consumer protection, consumers.

**Analysis**: Although there are also some strange words in Paraphrase 1, we can still capture the main meaning of it. Therefore, Paraphrase 1 can get a mediocre score in Fluency and a high score in Relevance. On the other hand, Paraphrase 1 has many lexical edits and turns the original sentence into two parallel sentences, so it can full marks in both terms of Lexical and Syntactical Diversity.

Paraphrase 2 is not really finished and repeats some words in the text (highlighted in blue), so it gets the lowest scores in Relevance and Fluency. Meanwhile, except for some incorrect word order transpositions, Paraphrase 2 is very similar to the original sentence. Therefore, it receives low scores in Lexical and Syntactical Diversity.

Obviously, Paraphrase 3 changes a lot lexically and syntactically. However, it is incoherent, difficult to understand (highlighted in red), so Paraphrase 3 scores high for Lexical and Syntactical Diversity and low for Fluency and Relevance.

Paraphrase 4 is a nonsensical text, which is not really finished and keeps repeating itself. Therefore, it gets the lowest scores from all aspects.

---

Table 20: Human evaluation questionnaire for paraphrase generation.