Diversifying Neural Text Generation with Part-of-Speech Guided Softmax and Sampling

Anonymous ACL submission

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

Context: The NK 2nd Division , concentrated in the Sinban-ni area west of the river , had , in effect , attacked straight east across the river and was trying to seize the two avenues of advance into Changnyong.
Text 1: They were joined by the 27th Battalion, US 24th Infantry Regiment, the 27th Regiment, and the 27th Regiment. The US 24th Infantry Division, under the command of Major General John R. Dempsey, was ordered to charge the US 24th Infantry Division, and capture it from the west.
Text 2: The NK 2nd Division, which had been involved in the assault, was forced to withdraw from the area north of the river. The NK 3rd Division, which had been fighting in the area since the beginning of the battle, was moved to the south. The NK 2nd Division, which had been fighting in the area since the start of the battle, had been pushed back to the south.
Text 3: The 2nd Division had been moving north from Alcester's position on the road, but were not expecting another attack. In the immediate aftermath of the attack, to keep the 2nd Division in reserve, which had been preparing for an attack on Hill 131. Along with the 3rd Battalion of the US 2nd Infantry Regiment, attacked Hill 129 at Pakchon on the way to Beaulieu.

Table 1: Examples of low-diversity generated text, given context from the Wikitext-103 dataset (Merity et al., 2017). Text 1 has a poor form diversity due to many useless repeating words (highlighted in blue). Text 2 keeps talking about only one single content, with similar lexicon and syntax (highlighted in orange), indicating low diversity in both terms of form and content. Though Text 3 has various syntactical and lexical forms with no repetition, all the content of it is about the "attacks", which means low content diversity. Text 1 is sampled from MLE, Text 2 from F^2-Softmax (Choi et al., 2020), and Text 3 from FACE (Jiang et al., 2019) (Section 5.1).

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 Sampling is proposed to promote text diversity effectively 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 decoding 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 methods have been widely used, and some studies propose 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 cumulative probability is at least α. However, those decoding-based methods are lack of controllability. 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 unlikelihood loss to implicitly reduce the frequent tokens and potential repeats. Other approaches, such as negative training (He and Glass, 2020), reinforcement learning (Shirai et al., 2020), and imitation learning (Zhou and Lampouras, 2020), have recently been applied to promote the diversity during the training phase. All above training-based methods 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 generation, 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 $p_\star(\mathbf{x})$ for all $\mathbf{x}$. Specifically, when the language model is a neural network, θ is regarded as the model parameters of the neural network, and we can factorize $p_\theta(\mathbf{x})$ as $p_\theta(\mathbf{x}) = \Pi_{t=1}^{T} p_\theta(x_t \mid \mathbf{x}_{<t})$. The conventional approach for learning the language model parameters θ is to maximize the log-likelihood by minimizing:

$$\begin{aligned} \mathcal{L}_{\text{MLE}}(\theta) &= -\sum_{t=1}^{T} \log p_\theta(x_t \mid \mathbf{x}_{<t}), \\ p_\theta(x_t \mid \mathbf{x}_{<t}) &= \frac{\exp \mathbf{h}_{t-1}^\top \mathbf{w}_{x_t}}{\sum_{x \in \mathcal{V}} \exp \mathbf{h}_{t-1}^\top \mathbf{w}_x}, \end{aligned} \tag{1}$$

where $\mathbf{h}_{t-1}$ 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

POSG 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 $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_T]$, where each ρ_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 ρ 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 ($\mathcal{V}_{\text{WP}}$) 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 $\mathcal{V}_{\text{IN}}$ 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 ρ_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 ρ_t is formulated as:

$$p_\theta(x_t, \rho_t \mid \mathbf{x}_{<t}) = p_{\theta_1}(\rho_t \mid \mathbf{x}_{<t}) \times p_{\theta_2}(x_t \mid \rho_t, \mathbf{x}_{<t}), \tag{2}$$

where $p_{\theta_1}(\rho_t \mid \mathbf{x}_{<t})$ is the next-POS probability and $p_{\theta_2}(x_t \mid \rho_t, \mathbf{x}_{<t})$ is the next-token probability conditioned on ρ_t. These probabilities are defined

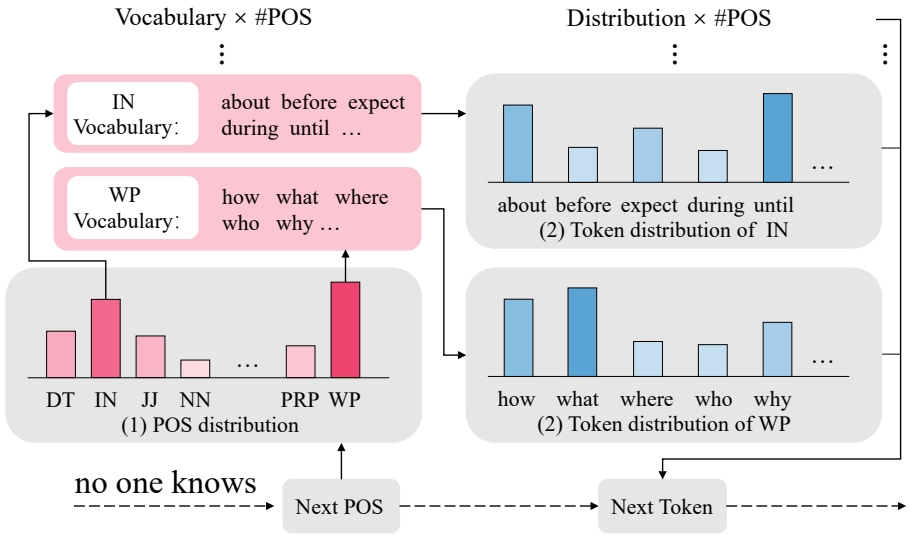

Figure 1: Illustration of POS Guided Softmax. The decoding process is decomposed into two stages: first predicts the next-POS distribution, and then decodes the next-token distribution from the vocabulary of the previously predicted POS. Since there exist some tokens with more than one POS, the final next-token distribution is the sum of all the POS's token distributions.

empirically by applying a linear output embedding on $\mathbf{h}_{t-1}$ and then a Softmax function respectively:

$$
\begin{aligned}
p_{\theta_1}\left(\rho_t \mid \mathbf{x}_{<t}\right) &= \frac{\exp \mathbf{h}_{t-1}^{\top} \mathbf{o}_{\rho_t}}{\Sigma_{\rho \in \mathcal{P}} \exp \mathbf{h}_{t-1}^{\top} \mathbf{o}_{\rho}}, \\
p_{\theta_2}\left(x_t \mid \rho_t, \mathbf{x}_{<t}\right) &= \begin{cases} \frac{\exp \mathbf{h}_{t-1}^{\top} \mathbf{w}_{x_t}}{\Sigma_{x \in \mathcal{V}_{\rho_t}} \exp \mathbf{h}_{t-1}^{\top} \mathbf{w}_{x}}, & \text{if } x_t \in \mathcal{V}_{\rho_t} \\ 0, & \text{otherwise} \end{cases},
\end{aligned}
\tag{3}
$$

where $\mathbf{o}_{\rho_t}$ and $\mathbf{w}_{x_t}$ are the output embeddings for $\rho_t \in \mathcal{P}$ and $x_t \in \mathcal{V}_{\rho_t}$, respectively. In this way, we regard 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: $p_\theta\left(x_t \mid \mathbf{x}_{<t}\right) = \sum_{\rho_t \in \mathcal{P}} p_\theta\left(x_t, \rho_t \mid \mathbf{x}_{<t}\right)$. Note that some tokens may have more than one POS, and $p_\theta\left(x_t, \rho_t \mid \mathbf{x}_{<t}\right) = 0$ 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:

$$
\begin{aligned}
\mathcal{L}_{\text{POS-Guided}}(\theta) = -\sum_{t=1}^{T} \big[& \log p_{\theta_1}\left(\rho_t \mid \mathbf{x}_{<t}\right) \\
& + \log p_{\theta_2}\left(x_t \mid \rho_t, \mathbf{x}_{<t}\right) \big].
\end{aligned}
\tag{4}
$$

4.2 POS Guided Sampling

We propose POS Guided Sampling based on POS Guided Softmax. Consistent with POS Guided Softmax, the key idea is to divide the whole sampling sample process into two stages: *POS 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:

$$
\begin{aligned}
p'_\theta\left(x_t \mid \mathbf{x}_{<t}\right) &= \sum_{\rho_t \in \mathcal{P}} \left[p'_{\theta_1}\left(\rho_t \mid \mathbf{x}_{<t}\right) \times p'_{\theta_2}\left(x_t \mid \rho_t, \mathbf{x}_{<t}\right) \right], \\
p'_{\theta_1}\left(\rho_t \mid \mathbf{x}_{<t}\right) &= \begin{cases} \frac{p_{\theta_1}\left(\rho_t \mid \mathbf{x}_{<t}\right)}{\mathcal{Z}_{\theta_1}}, & \text{if } \rho_t \in \mathcal{P}' \\ 0, & \text{otherwise} \end{cases}, \\
p'_{\theta_2}\left(x_t \mid \rho_t, \mathbf{x}_{<t}\right) &= \begin{cases} \frac{p_{\theta_2}\left(x_t \mid \rho_t, \mathbf{x}_{<t}\right)}{\mathcal{Z}_{\theta_2}}, & \text{if } x_t \in \mathcal{V}'_{\rho_t} \\ 0, & \text{otherwise} \end{cases}, \\
\mathcal{Z}_{\theta_1} &= \sum_{\rho \in \mathcal{P}'} p_{\theta_1}\left(\rho \mid \mathbf{x}_{<t}\right), \\
\mathcal{Z}_{\theta_2} &= \sum_{x \in \mathcal{V}'_{\rho_t}} p_{\theta_2}\left(x \mid \rho_t, \mathbf{x}_{<t}\right),
\end{aligned}
\tag{5}
$$

where $\mathcal{P}' \subset \mathcal{P}$ is a POS set containing top-k most probable POS tags, and $\mathcal{V}'_{\rho_t} \subset \mathcal{V}_{\rho_t}$ is the smallest token set such that $\sum_{x \in \mathcal{V}'_{\rho_t}} p_{\theta_2}\left(x \mid \rho_t, \mathbf{x}_{<t}\right) \geq \alpha^{(\text{token})}$. $k^{(\text{POS})}$ and $\alpha^{(\text{token})}$ $(0 < \alpha^{(\text{token})} \leq 1)$ 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.

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 (F^2-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[1] 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[2] 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 $\alpha^{(\text{token})} = 0.5$ 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 $\mathcal{P}$ is much smaller than the total token vocabulary. We then conduct a grid search to find the $k^{(\text{POS})}$ whose generated continuations have the smallest reverse language model score (Semeniuta et al., 2018) on the validation set. $k^{(\text{POS})}$ 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 (Montahaei et 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-

[1] https://s3.amazonaws.com/research.metamind.io/wikitext/wikitext-103-v1.zip

[2] The XPOS tags are language-specific part-of-speech tags from the Universal Dependency Treebanks.

Models	Self-BLEU4 ↓	Rep ↓	Uniq ↑	Distinct ↑			PPL ↓	KLD ↓	MS-Jaccard ↑		
				n=1	n=2	n=3			n=1	n=2	n=3
MLE	46.9	1.86	11.7k	50.2	77.2	86.2	**32.7**	1.34	56.9	38.2	25.4
FACE	34.2	1.56	**14.9k**	60.0	85.1	90.6	36.1	1.18	58.6	37.6	24.0
F^2-Softmax	51.5	4.09	10.8k	42.4	65.3	75.2	35.0	1.58	51.5	33.7	22.4
UL	42.4	0.240	12.8k	**61.2**	87.8	93.3	37.0	1.20	61.2	40.4	**26.2**
MoS	55.3	3.99	8.40k	48.2	74.3	83.0	38.2	1.48	56.9	38.1	25.4
POSG	**34.1**	**0.000**	13.8k	60.2	**88.8**	**94.3**	34.4	**1.17**	**62.2**	**40.7**	25.9

Table 2: Automatic evaluation results for different models on the language modeling task. Numbers $n \in \{1, 2, 3\}$ in the column heads under Distinct and MS-Jaccard refer to n-gram. (**Bold**: the best; Underline: the second best).

Models	Distinct ↑					
	1-P	1-G	2-P	2-G	3-P	3-G
MLE	16.0	39.2	38.5	61.0	54.7	70.8
FACE	17.8	49.8	46.6	73.0	65.4	80.8
F^2-Softmax	16.4	41.1	39.3	63.8	55.8	74.0
UL	18.0	51.7	46.5	76.8	66.3	85.2
MoS	16.4	41.1	40.0	63.8	56.5	72.9
POSG	19.9	56.2	58.1	85.3	80.6	92.3
Human	21.7	67.7	61.8	93.0	83.8	95.9
PPMCC	0.988		0.986		0.986	

Table 3: 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.

Models	Div. ↑	Qua. ↑
MLE	2.86*	3.10
FACE	3.32*	3.18
F^2-Softmax	2.35*	2.80*
UL	3.36*	**3.20**
MoS	2.79*	3.06*
POSG	**3.45**	3.17

Table 4: Human evaluation on language modeling. * denotes statistical significance compared with POSG (Mann-Whitney u-test, $p < 0.1$).

tain comparable generation quality.

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. In terms of distinct n-POS, POSG also surpasses all the baselines. This demonstrates that our proposed model can substantially promote the POS diversity. Moreover, the Pearson correlations between distinct n-POS and n-gram are extremely high, which indicates that the high POS diversity indeed leads to the high text diversity.

Human evaluation For the language modeling task, following Tevet and Berant (2021) we randomly sample 100 generated continuations from each model. Each of them is scored between 1 to 5 (5 is the best), by five workers to evaluate the overall *Diversity* (Div.) and *Quality* (Qua.). The results of the human evaluation on language modeling are shown in Table 4. It can be seen that our POSG significantly outperforms all other baselines in diversity, and performs relatively well in quality.

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 $\max(3, H_{\max} - 4)$ to reduce the impact from exemplar sentence, where $H_{\max}$ 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 corrspond exemplar sentences for inference. For fair comparison, we closely follow Kumar et al. (2020)

[3] https://drive.google.com/file/d/1rbF3daJjCsa1-fu2GANeJd2FBXos1ugD/view

[4] https://github.com/malllabiisc/SGCP

Models	Self-WER↑	Self-BLEU4↓	Distinct↑			BERTScore↑	BLEU4↑	ROUGE↑		
			n=1	n=2	n=3			1	2	L
MLE	74.2	25.1	78.4	82.8	78.5	47.4	9.81	38.1	16.9	38.8
FACE	73.0	25.0	78.9	83.6	79.6	48.1	10.1	38.7	17.2	39.1
F2-Softmax	76.4	28.0	78.2	83.0	79.5	**53.9**	**11.4**	41.1	**19.5**	**42.8**
UL	77.2	21.2	80.1	**85.3**	80.9	36.0	7.59	30.6	13.3	30.3
SGCP	83.0	28.6	81.9	82.6	77.7	47.9	9.91	**41.3**	17.7	41.1
POSG	**89.7**	**19.6**	**82.1**	**85.3**	**81.8**	48.3	9.79	40.3	17.1	39.4

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).

Models	Div. ↑		Flu. ↑	Rel. ↑
	Lex.	Syn.		
MLE	2.92	2.65⋆	3.34⋆	3.09⋆
FACE	2.91	2.58⋆	**3.60**	3.35
F^2-Softmax	2.77⋆	2.57⋆	3.59	**3.38**
UL	3.00	2.68⋆	3.37⋆	3.17⋆
SGCP	2.74⋆	2.67⋆	3.50	3.21⋆
POSG	**3.02**	**2.79**	3.58	3.35

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

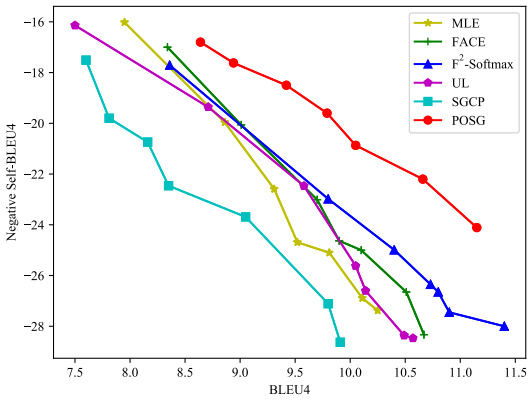

Figure 2: Quality-diversity trade-off for different models on paraphrase generation. The x-axis measures BLEU4 for quality, and the y-axis measures negative Self-BLEU4 for diversity. Both are the bigger the better.

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 (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 baselines 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., F^2-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:

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

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.

Models	Self-WER↑	Self-BLEU4↓	Distinct↑			BERTScore↑	BLEU4↑	ROUGE↑		
			n=1	n=2	n=3			1	2	L
POSG	**89.7**	**19.6**	**82.1**	**85.3**	**81.8**	48.3	9.79	40.3	17.1	39.4
w/o POSG-Sampling	87.6	24.1	78.0	80.5	78.5	**52.6**	**11.1**	**40.9**	**19.7**	**42.4**
MLE	74.2	25.1	78.4	82.8	78.5	47.4	9.81	38.1	16.9	38.8

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

Adjective Probability	Adjs. per Sentence	Self-BLEU4↓	BLEU4↑
$\times 0.1$	0.43	20.2	9.47
$\times 1$	0.66	19.6	9.79
$\times 10$	1.04	18.7	9.45

Table 9: Results of controllability analysis on the paraphrase generation task. "$\times n$" means that we manually multiply the probability of "Adjective" by n.

Lexical Diversity (LeD.), and *Syntactical Diversity* (SyD.), *Fluency* (Flu.), *Relevance* (Rel.). All these aspects are scored between 1 to 5, the higher the better. As shown in Table 6, the results of the human evaluation are strongly consistent with the automatic evaluation. Compared with MLE, UL and SGCP, POSG substantially improves the generation quality, and it only has a tiny gap from the best model in fluency and relevance scores. Meanwhile, POSG has the best scores in diversity, which further verifies that our proposed methods can generate more lexically and syntactically diverse paraphrases. The detailed questionnaire, the inter-annotator agreement, and other details are shown in Appendix E.

5.4 Ablation Study

We perform ablation studies to reveal the effect of POS Guided Softmax and POS Guided Sampling. As shown in Table 7 and Table 8, compared with MLE, POS Guided Softmax (without POS Guided Sampling) can improve text quality for both the tasks. Moreover, POS Guided Sampling can dramatically promote text diversity for both the tasks. These results confirm the effectiveness of both the components.

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

Emanuele Bugliarello and Desmond Elliott. 2021. The role of syntactic planning in compositional image captioning. In *In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 593–607.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. Language gans falling short. In *In Proceedings of the 8th International Conference on Learning Representations*.

Yue Cao and Xiaojun Wan. 2020. Divgan: Towards diverse paraphrase generation via diversified generative adversarial network. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2411–2421.

Byung-Ju Choi, Jimin Hong, David Keetae Park, and Sang Wan Lee. 2020. Fˆ2-softmax: Diversifying neural text generation via frequency factorized softmax. In *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9167–9182.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David A. Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 10695–10704.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 889–898.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2020. A theoretical analysis of the repetition problem in text generation. *arXiv preprint arXiv:2012.14660*.

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

Tianxing He and James R. Glass. 2020. Negative training for neural dialogue response generation. In *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2044–2058.

Xinwei He, Baoguang Shi, Xiang Bai, Gui-Song Xia, Zhaoxiang Zhang, and Weisheng Dong. 2019. Image caption generation with part of speech guidance. *Pattern Recognit Letters*, 119:229–237.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *In Proceedings of the 8th International Conference on Learning Representations*.

Ryan Hornby, Clark Taylor, and Jungyeul Park. 2017. Corpus selection approaches for multilingual parsing from raw text to universal dependencies. In *In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 198–206.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. In *In Proceedings of the World Wide Web Conference*, pages 2879–2885.

Ilya Kulikov, Alexander H. Miller, Kyunghyun Cho, and Jason Weston. 2018. Importance of a search strategy in neural dialogue modelling. *arXiv preprint arXiv:1811.00907*.

Solomon Kullback. 1997. *Information theory and statistics*. Courier Corporation.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha P. Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Trans. Assoc. Comput. Linguistics*, 8:330–345.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics System Demonstrations*, pages 55–60.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *In Proceedings of 5th International Conference on Learning Representations*.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *In Proceedings of the 29th International Conference on Machine Learning*.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. *arXiv preprint arXiv:1904.03971*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. 2018. On accurate evaluation of gans for language generation. *CoRR*, abs/1806.04936.

Rico Sennrich and Barry Haddow. 2016a. Linguistic input features improve neural machine translation. In *In Proceedings of the First Conference on Machine Translation*, pages 83–91.

Rico Sennrich and Barry Haddow. 2016b. Linguistic input features improve neural machine translation. In *In Proceedings of the First Conference on Machine Translation*, pages 83–91.

Keisuke Shirai, Kazuma Hashimoto, Akiko Eriguchi, Takashi Ninomiya, and Shinsuke Mori. 2020. Neural text generation with artificial negative examples. *arXiv preprint arXiv:2012.14124*.

Shang-Yu Su, Kai-Ling Lo, Yi Ting Yeh, and Yun-Nung Chen. 2018. Natural language generation by hierarchical decoding with linguistic patterns. In *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 61–66.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *In Advances in Neural Information Processing Systems*, pages 5998–6008.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7371–7379.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *In Proceedings of the 8th International Conference on Learning Representations*.

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

Michael Wray, Gabriela Csurka, Diane Larlus, and Dima Damen. 2019. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *In the 2019 IEEE/CVF International Conference on Computer Vision*, pages 450–459.

Kexin Yang, Wenqiang Lei, Dayiheng Liu, Weizhen Qi, and Jiancheng Lv. 2021. Pos-constrained parallel decoding for non-autoregressive generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5990–6000. Association for Computational Linguistics.

Xuewen Yang, Yingru Liu, Dongliang Xie, Xin Wang, and Niranjan Balasubramanian. 2019. Latent part-of-speech sequences for neural machine translation. In *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 780–790.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *In Proceedings of the 8th International Conference on Learning Representations*.

Giulio Zhou and Gerasimos Lampouras. 2020. Generating safe diversity in nlg via imitation learning. *arXiv preprint arXiv:2004.14364*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

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.

	Train	Valid	Test
#Articles	28,475	60	60
#Tokens	103,227,021	217,646	245,569

Table 10: Statistics of Wikitext-103.

	Train	Valid	Test
#Sentence	1,640,709	3,000	800

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.

Hyperparameters	Wikitext-103	ParaNMT-50M
Vocabulary size	267,735	100,000
Batch size	12	96
Learning rate	0.0001	0.0001
Finetuning LR	0.00001	0.00001
Finetuning step	1500	1500
Gradient clipping	0.25	0.25
Weight decay	0.001	0.001
Droupout	0.1	0.1
Optimizer	Adam	Adam
-β_1	0.9	0.9
-β_2	0.999	0.999
-ϵ	1e-8	1e-8

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 finetuning with corresponding fineturning LR and fineturning 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 $H(p)$: $H(p) = -\sum_{x\in\mathcal{V}} p(x)\log p(x)$. 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 $-p(x_{\max})\log p(x_{\max})$ to $-\sum_{x\in\mathcal{V}_k}\frac{p(x)}{Z_k}\log\frac{p(x)}{Z_k}$, where $x_{\max}$ is the token with the max probability, $\mathcal{V}_k$ is the set of top-k most probable tokens, $Z_k = \sum_{x\in\mathcal{V}_k} p(x)$, and obviously $x_{\max} \in \mathcal{V}_k$.

Prefix: Below them, North Koreans continued crossing the river and moving supplies forward to their combat units, some of them already several miles eastward. The North Koreans quickly discovered Task Force Manchu group. They first attacked it at 14:00 that afternoon, and were repulsed
MLE: by the North Koreans. On the morning of September 8, the North Korean forces launched a surprise attack on the high ground west of the Kum River. At 16:30, the North Korean force launched a massive attack on the North Korean force, but the initial attack was not successful. The North Korean offensive was halted by the remnants of the North Korean forces
FACE: by heavy machine-gun fire. In the early morning hours of 8 September, North Korean troops were alerted to attack on the perimeter. On 9 September, a force of 20,000 men led by Lieutenant Colonel Robert E. telluride began to attack the North Korean lines, suffering little damage. By 14:00 on 9 September, North Korean forces had crossed the Naktong River just before midnight.
F^2-Softmax: by the North Koreans. The North Koreans were ordered to withdraw to the rear of the North Koreans. They then launched a frontal attack on the south side of the river. The North Koreans then launched a frontal attack on the North Korean right flank. The North Korean right flank was soon overrun by the North Koreans. The North Koreans were subsequently repulsed by the North Koreans,
UL: by North Korean fire, which forced the North Koreans to retreat. A further assault by the 1st U.S. Infantry Regiment followed in the afternoon, and after seven hours of fighting, the 2nd U.S. Infantry Regiment broke off the attack and retreated across the river. The survivors of the Battle of tellers managed to escape to a new bridge. Task Force presaged, but by 20:00 the North Koreans were completely surrounded by North Korean troops.
MoS: by the 9th Infantry Regiment. At 17 : 00, the North Koreans took the road from the Korean border to the north, and began firing on the northern flank of the North Korean forces. The North Koreans then withdrew to the northern flank of the Korean army, where they advanced into the river and quickly attacked the North Koreans. At 16 : 00, the North Koreans began firing on the North Koreans, and a number of North Korean soldiers, including the 5th Cavalry Regiment, attacked the North Koreans.
POSG: by the North Koreans, beginning their advance south of Osan on 18 September. By nightfall on 24 September, Ho Chi Minh had secured its flank, while the South Koreans had captured the town of Phong on the west of Taejon. The North Koreans had retreated to Pyongtaek, and in the afternoon of 22 September two North Koreans were killed there, leaving behind the town to the survivors.

Table 13: Examples of language modeling on Wikitext-103 dataset. Repeating text is highlighted in blue, dull text with single context is highlighted in orange, and incoherent text is highlighted in red.

Now, we prove that our POSG with two sampling stages can lead to the entropy increasing, compared with one-stage top-k sampling as an example.

For one-stage top-k sampling,

$$
\begin{aligned}
H\left(p\right)^{\text{(top-k)}} &= -\sum_{x\in\mathcal{V}_k}\frac{p(x)}{Z_k}\log\frac{p(x)}{Z_k} \\
&= -\sum_{x\in\mathcal{V}_k}\frac{\sum_{\rho\in\mathcal{P}}p(x,\rho)}{Z_k}\log\frac{\sum_{\rho\in\mathcal{P}}p(x,\rho)}{Z_k} \\
&= -\log|\mathcal{P}| - \sum_{x\in\mathcal{V}_k}\frac{\sum_{\rho\in\mathcal{P}}p(x,\rho)}{Z_k}\log\frac{\sum_{\rho\in\mathcal{P}}p(x,\rho)}{Z_k\times|\mathcal{P}|}
\end{aligned}
\tag{6}
$$

According to the Log sum inequality, it follows:

$$
\begin{aligned}
H(p)^{\text{(top-k)}} &\geq -\log|\mathcal{P}| - \sum_{x\in\mathcal{V}_k}\sum_{\rho\in\mathcal{P}}\frac{p(x,\rho)}{Z_k}\log\frac{p(x,\rho)}{Z_k} \\
&= -\log|\mathcal{P}| - \sum_{\rho\in\mathcal{P}}\sum_{x\in\mathcal{V}_k}\frac{p(x,\rho)}{Z_k}\log\frac{p(x,\rho)}{Z_k}
\end{aligned}
\tag{7}
$$

Since $p(x,\rho)=0$ for $x\notin\mathcal{V}_\rho$, it follows:

$$
H(p)^{\text{(top-k)}} \geq -\log|\mathcal{P}| - \sum_{\rho\in\mathcal{P}}\sum_{x\in\mathcal{V}_{k,\rho}}\frac{p(x,\rho)}{Z_k}\log\frac{p(x,\rho)}{Z_k} \tag{8}
$$

where $\mathcal{V}_{k,\rho}=\{x\in\mathcal{V}_k \mid \rho\in\text{POS}(x)\}$, $\text{POS}(x)$ is the set of all POS tags of token x. Thus, $\mathcal{V}_{k,\rho}\subseteq\mathcal{V}_k$.

For our POSG with two sampling stages,

$$
\begin{aligned}
H(p)^{\text{(POS)}} &= -\sum_{x\in\mathcal{V}}\sum_{\rho\in\mathcal{P}}p'(x,\rho)\log\sum_{\rho\in\mathcal{P}}p'(x,\rho) \\
&= -\sum_{x\in\mathcal{V}}\sum_{\rho\in\mathcal{P}}p'(\rho)p'(x\mid\rho)\log\sum_{\rho\in\mathcal{P}}p'(\rho)p'(x\mid\rho)
\end{aligned}
\tag{9}
$$

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

$$
\begin{aligned}
H(p)^{\text{(POS)}} &\geq -\log|\mathcal{P}| \\
&- \sum_{x\in\mathcal{V}}\sum_{\rho\in\mathcal{P}}p'(\rho)p'(x\mid\rho)\log p'(\rho)p'(x\mid\rho)
\end{aligned}
\tag{10}
$$

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

$$
p'(x\mid\rho)=\begin{cases}\frac{p(x|\rho)}{Z_2}, & \text{if } x\in\mathcal{V}_{\rho,k}\\ 0, & \text{otherwise}\end{cases}, Z_2=\sum_{x\in\mathcal{V}_{\rho,k}}p(x)
$$

Note that, in our paper, we denote all the tokens whose POS is ρ as a vocabulary $\mathcal{V}_\rho$, and here, $\mathcal{V}_{\rho,k}$

Models	Self-WER↑	Self-BLEU4↓	Distinct↑			BERTScore↑	BLEU4↑	ROUGE↑		
			n=1	n=2	n=3			1	2	L
top-k	100.8	13.6	86.9	88.9	83.7	39.4	6.49	33.5	12.1	32.3
POSG	102.1	13.6	86.9	88.2	83.4	**43.3**	**7.71**	**36.4**	**14.1**	**34.9**
Δ	+1.3	+0.0	+0.0	-0.7	-0.3	**+3.9**	**+1.22**	**+2.9**	**+2.0**	**+2.6**

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 able to economic point of the city .
SGCP: this will make economic sense for the city .
POSG: it is what makes good economic sense to the city .

Table 15: Examples of paraphrase generation on ParaNMT-50M.

Input Sentence:	he (PRP) was (VBD) smiling (VBG) , clearly (RB) **delighted (JJ)**
$\times 0.1$	he (PRP) was (VBD) smiling (VBG) , and (CC) he (PRP) was (VBD) clearly (RB) pleased (VBN) with (IN) joy (NN)
$\times 1$	he (PRP) was (VBD) smiling (VBG) and (CC) apparently (RB) **delighted (JJ)** with (IN) joy (NN) in (IN) his (PRP$) face (NN)
$\times 10$	he (PRP) was (VBD) still (RB) smiling (VBG) and (CC) **delighted (JJ)** with (IN) **apparent (JJ)** joy (NN) in (IN) his (PRP$) face (NN)

Table 16: Examples of controllability analysis on the paraphrase generation task.

is the set of top-k most probable tokens in $\mathcal{V}_\rho$. Thus, $\mathcal{V}_{\rho,k} \subseteq \mathcal{V}_\rho$. Then, it follows:

$$
\begin{aligned}
&H(p)^{(\text{POS})} \geq -\log|\mathcal{P}| \\
&-\sum_{x\in\mathcal{V}}\sum_{\rho\in\mathcal{P}} p(\rho)p'(x\mid\rho)\log p(\rho)p'(x\mid\rho) \\
&= -\log|\mathcal{P}| - \sum_{\rho\in\mathcal{P}}\sum_{x\in\mathcal{V}_{\rho,k}} p(\rho)\frac{p(x\mid\rho)}{Z_2}\log p(\rho)\frac{p(x\mid\rho)}{Z_2} \\
&= -\log|\mathcal{P}| - \sum_{\rho\in\mathcal{P}}\sum_{x\in\mathcal{V}_{\rho,k}} \frac{p(x,\rho)}{Z_2}\log\frac{p(x,\rho)}{Z_2}
\end{aligned}
\tag{11}
$$

Since $\mathcal{V}_{k,\rho} \subseteq \mathcal{V}_{\rho,k}$ 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 $H(p)^{(\text{POS})}$ is greater than or equal to the lower bound of $H(p)^{(\text{top-k})}$. 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 Additional Analysis

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 $k^{(\text{POS})} = 5$ in POS sampling, $k^{(\text{token})} = 500$ 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-

	Div.	Qua.
Krippendorff's α	0.57	0.40

Table 17: Agreement analysis for annotators labels on the language modeling task.

Models	Div.		Flu.	Rel.
	Lex.	Syn.		
Krippendorff's α	0.54	0.37	0.71	0.63

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

fer from a severe repetition problem, and they also generate many similar sentences about a single content. Due to the low-diversity problem, MoS even generates some illogical text, such as "the North Koreans began firing on the North Koreans". FACE produces a large amount of incoherent text, making the text somewhat hard to read. UL and our POSG alleviate those problems, while our model performs relatively better.

Additionally, examples of paraphrase generation are shown in Table 15. We observe that almost all models can generate high-quality paraphrases with well-preserved semantic meanings, while our POSG exhibits more syntactic diversity than other baselines.

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 ($0.2 \leq \kappa \leq 0.4$) or moderate agreement ($0.4 \leq \kappa \leq 0.6$).

F Impact Statement

Our work has developed generic generation methods to promote text diversity while maintaining comparable quality. Therefore, despite the contributes 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.

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. . .

Continuation 1: Score: Diversity 3, Quality 3

second action around 380 yards <unk> 290 m <unk> 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 <unk> 16 km <unk> from Highton .

Continuation 2: Score: Diversity 1, Quality 5

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

Continuation 3: Score: Diversity 1, Quality 1

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

Continuation 4: Score: Diversity 4, Quality 3

" 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

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.