

Improving Diversity of Neural Text Generation via Inverse Probability Weighting

Xinran Zhang¹, Maosong Sun^{12*}, Jiafeng Liu¹, Xiaobing Li¹

¹Department of Music Artificial Intelligence and Music Information Technology
Central Conservatory of Music, Beijing, China

²Department of Computer Science and Technology, Tsinghua University, Beijing, China
Institute for Artificial Intelligence, Tsinghua University, Beijing, China
State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
zhangxr.wspn@gmail.com, sms@tsinghua.edu.cn

Abstract

The neural network based text generation suffers from the text degeneration issue such as repetition. Although top- k sampling and nucleus sampling outperform beam search based decoding methods, they only focus on truncating the “tail” of the distribution and do not address the “head” part, which we show might contain tedious or even repetitive candidates with high probability that lead to repetition loops. They also do not fully address the issue that human text does not always favor high probability words. To explore improved diversity for text generation, we propose a heuristic sampling method inspired by inverse probability weighting. We propose to use interquartile range of the predicted distribution to determine the “head” part, then permute and rescale the “head” with inverse probability. This aims at decreasing the probability for the tedious and possibly repetitive candidates with higher probability, and increasing the probability for the rational but more surprising candidates with lower probability. The proposed algorithm provides a controllable variation on the predicted distribution which enhances diversity without compromising rationality of the distribution. We use pre-trained language model to compare our algorithm with nucleus sampling. Results show that our algorithm can effectively increase the diversity of generated samples while achieving close resemblance to human text.

1 Introduction

With the fast development of deep learning technologies, many natural language processing (NLP) tasks have witnessed significant improvements of performance. The Transformer architecture proposed by Vaswani et al. (2017) and its subsequential representative variants, including BERT by Devlin et al. (2019), GPT-2 by Radford et al. (2019),

Transformer XL by Dai et al. (2019), XLNET by Yang et al. (2019) and the recent GPT-3 by Brown et al. (2020) have opened a new paradigm for NLP tasks. Text generation, one of the vital NLP tasks and also a typical case of the classical sequence modeling problem in NLP, has attracted numerous research attention in recent years, and benefits a lot from this new paradigm. However, there still exists some unsolved difficult issues for text generation, among which the most difficult one is perhaps the text degeneration issue (Holtzman et al., 2020), that is, the texts generated by the aforementioned neural methods exhibit a strong tendency to be repetitive with low diversity.

Recent work by Kang and Hashimoto (2020) has reveals the brittleness of maximum likelihood loss that is commonly used in language model training, which will render a sub-optimal and unreliable distribution. Consequently, directly sampling on such distribution will produce unsatisfactory results. Kang and Hashimoto (2020) propose to truncate loss (during training) to enhance reliability of the predicted distribution. On the other hand, stochastic sampling methods with vocabulary filtering such as top- k sampling (Fan et al., 2018; Holtzman et al., 2018) and nucleus sampling (top- p sampling, Holtzman et al., 2020) truncate the predicted distribution (during decoding), filter a top portion of vocabulary and exclude unreliable candidates with low probability, which achieve better results than pure sampling (directly sampling on the distribution) or beam search based decoding methods. However, these methods only focus on truncating the unreliable “tail” of the distribution, and do not fully address the issue that human text does not always favor high probability candidates (Holtzman et al., 2020), i.e., the “head” of the distribution remains unprocessed. We show that nucleus sampling still tends to generate tedious and even repetitive samples with low diversity caused by the “head” part. To explore improved diversity for text

*Corresponding author.

generation, we take a step forward on the basis of stochastic sampling with vocabulary filtering. We propose the interquartile range inverse probability (IQR-IP) sampling algorithm. It brings a controllable permutation on the “head” part of the predicted distribution on the filtered vocabulary to enhance diversity while still preserving the rationality of the distribution. For evaluation, we use pre-trained GPT-2 model by Radford et al. (2019) to generate samples with different sampling methods. Results show that our algorithm can increase diversity while maintaining close resemblance to human text compared with traditional methods.

2 Stochastic Sampling with Vocabulary Filtering for Text Generation

2.1 Top- k Sampling and Nucleus Sampling

Typical stochastic sampling for text generation starts by truncating the “tail” of the predicted distribution and filtering a top portion from the original vocabulary (denoted by V) according to some metric, then performs stochastic sampling according to the regularized distribution on the filtered vocabulary. For example, the top- k sampling (Fan et al., 2018; Holtzman et al., 2018) filters the top k probable candidates as follows.

$$V^k = \{x \mid \text{rank}(p(x)) \leq k, x \in V\}, \quad (1)$$

where $p(x)$ denotes the predicted distribution of the model, and rank refers to the ranking order of $p(x)$. The auto-regressive dependency of $p(x)$ on the context of word x on each sampling step is omitted for simplicity throughout this work. Another common choice is the nucleus sampling (top- p sampling, Holtzman et al., 2020) which filters the vocabulary with top p mass of cumulative probability as follows.

$$V^p = \{x \mid \text{cdf}(x) \leq p, x \in V\}, \quad (2)$$

where the cumulative density function $\text{cdf}(x)$ is calculated on the sorted distribution of $p(x)$ (in descending order is implied throughout this work).

2.2 Traditional Methods: Truncating the Unreliable “Tail”

According to Holtzman et al. (2020), top- k sampling and nucleus sampling can generate much better samples than beam search based decoding method (Li et al., 2016; Shen et al., 2017; Wiseman

et al., 2017) for text generation, because the latter always chooses candidates with high probability, generating samples that have much lower perplexity compared to human text and tend to be tedious with low diversity. On the other hand, directly performing stochastic sampling on the unfiltered predicted distribution (referred to as “pure sampling” by Holtzman et al., 2020) will produce samples with poor quality, since the sampling process will occasionally choose too low probability candidates that might be unreasonable. As a straightforward method, top- k sampling and nucleus sampling filters a top portion of the vocabulary in order to exclude the unreliable “tail” of the predicted distribution during decoding phase. Recent work by Kang and Hashimoto (2020) reveals the fragility of log loss training and adopts a new filtering method which excludes unreliable training samples with outlier loss (higher than a quantile of loss calculated from latest training batches) during training phase. This is essentially similar to the idea of filtering on vocabulary that truncates the unreliable “tail” with low probability.

2.3 Repetition Loops Caused by “Head”

However, these filtering methods only focus on truncating the “tail” of the distribution, and do not address the “head” part, which we show may lead to the annoying repetition issue. To explore the behavior of repetition loops, we use GPT-2 Small (Radford et al., 2019) to generate 5000 samples with the same input context and set maximum generation length to be 1024 (i.e., the maximum context length of GPT-2 model). The sharing input context is “She walks in beauty” (from Lord Byron’s most famous poetry). We use Equation 12 in Section 4.4 to detect repetition loops (using $H_{rep} < 2$ for all 200-length token windows) and present the trajectory of first 3 generated loops of a specific sample that contains infinite loops of “She walks in beauty.” (with generated period) in Figure 1. We found several phenomena that cause this repetition. First, the repetitive candidates always have high probability and high rank in the predicted distribution (see “*” labeled candidates in each heatmap box). Second, the repetition tendency grows stronger when more loops occur (due to a few sampling steps that happen to pick repetitive token in non-extreme distribution, e.g., in Loop #2), as the flat distribution in Loop #1 (e.g., “She” and “walks”) gradually becomes peaked distribu-

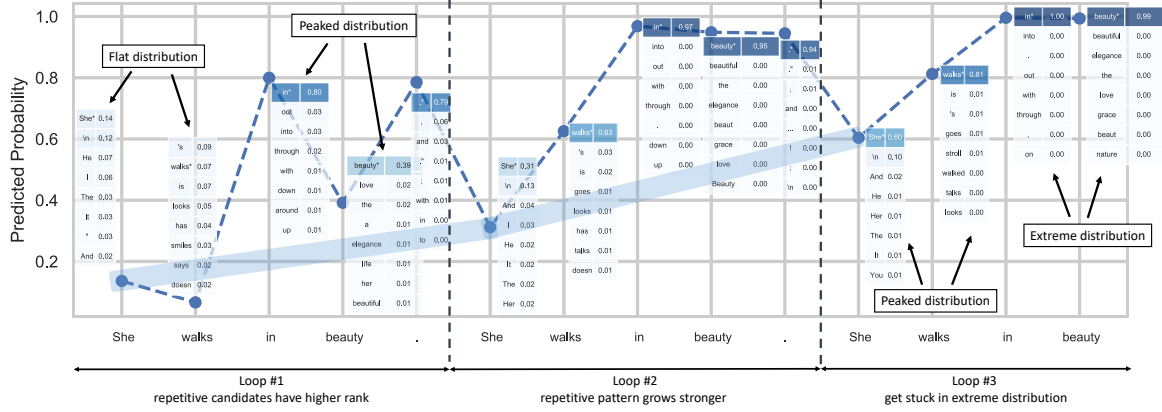


Figure 1: Trajectory of predicted probability (“o” marker) and predicted distribution (heatmap box besides each marker in “word-probability” format, with the sampled word marked by “*”) for first 3 repetition loops. It is one of the 5000 samples decoded by top- p sampling ($p=0.95$) with “She walks in beauty” as input context using GPT-2 Small (Radford et al., 2019). This specific sample contains infinite repetitive loops of “She walks in beauty.” (with generated period). The trajectory of repetitive word “She” is highlighted in shadow which shows the increase of predicted probability and the gradually peaked predicted distribution.

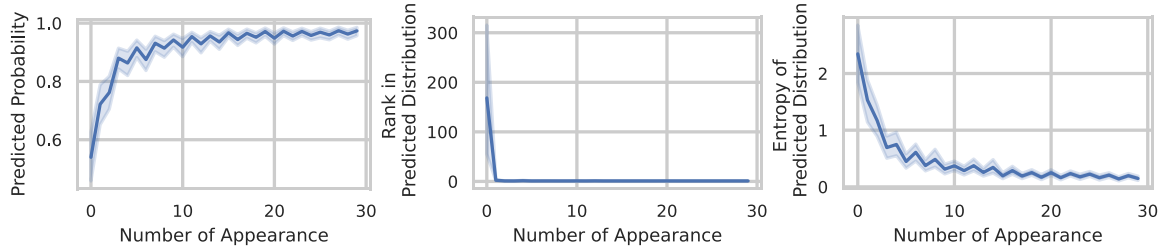


Figure 2: Trajectories of repetitive candidates extracted from samples that contain repetition loops. Repetition loops are detected using $H_{rep} < 2$ on 200-length of token window. Repetitive candidates that appears more than 30 times in the window are extracted and aligned to form their trajectories. It shows that a few appearances of repetitive candidates quickly lead model to extreme distribution that causes repetition loops.

tion in Loop #3, and peaked distribution in Loop #1 (e.g., “in” and “beauty”) becomes extreme distribution in Loop #3, which reciprocally contributes to stronger repetition pattern in the context. Third, the predicted distribution got stuck in an extreme case that assigns almost all probability mass for repetitive candidates (e.g., “in” and “beauty” in Loop #3).

To further verify these phenomena, we extract the trajectories of all repetitive candidates from samples with repetition loops (e.g., aligning all appearances of “She” in a sample to form its trajectory). Figure 2 presents the trajectories of predicted probability, rank in predicted distribution and entropy of predicted distribution, where x axis is the number of appearance of repetitive candidates. It shows that after a few appearances of repetitive candidates, the predicted distribution will quickly get stuck in extreme cases where predicted probability approaches 1, rank approaches 1, and en-

trophy approaches 0, which will surely render repetition loops. From these results, it is clear that the model tends to predict high probability for repetitive candidates that exist in the context. This is in accordance with analysis by Kang and Hashimoto (2020), which shows that words directly entailed in the context tend to have lower loss, i.e., higher predicted probability. This will lead the model to generate samples that might contain repetition loops with low quality.

2.4 Improving Diversity by Emphasizing on Less Probable Candidates in “Head”

On deeper thoughts, recall the results by Holtzman et al. (2020) which show that human text does not always choose high probability candidates, as the beam search based decoding method that generates samples with low perplexity actually deviates from human text behavior (see Figure 2, Holtzman et al., 2020). This is in accordance with human in-

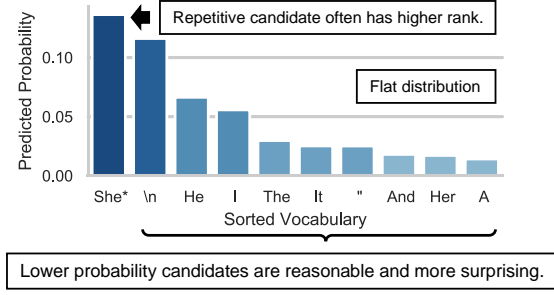


Figure 3: “Head” part of predicted distribution on the first sampling step of Loop #1 from Figure 1. Besides “She” that has highest predicted probability, lower probability candidates (“\n”, “He”, “I”, “The”, ...) are also reasonable choices with higher diversity. If using inverse probability weighting to emphasize on these candidates, the fluency of samples will not be compromised, while repetition will be suppressed and diversity will be improved.

stinct that always composing high-frequency words with high probability will be tedious and less surprising, while choosing reasonable words with low probability actually helps to compose better and diverse passages. As a result, the repetition loops phenomenon can be regarded as a representative of low diversity issue caused by always picking high probability candidates. As is shown in Figure 3, lower probability candidates in a flat distribution are actually reasonable but more surprising. Consequently, it is possible to increase diversity by emphasizing on these “less probable” candidates while not compromising the rationality of the distribution as well as the fluency of the samples. Intuitively, this can be achieved similarly to the inverse probability weighting technique that is commonly seen in causal inference (see Chapter 2, [Hernán MA, Robins JM, 2020](#)). Inspired by this, as long as we can identify a small subset of candidates in the “head” part of the distribution that contains all reasonable candidates (such as in Figure 3), we may use inverse probability weighting to rescale the distribution for these candidates to suppress repetition and increase diversity.

On a more generalized analysis for diversity, traditional vocabulary filtering methods achieve an “adversarial” balancing between rationality and diversity, where one side of diversity is to keep all vocabulary without any truncation (like pure sampling), and the opposite side of rationality is to keep the top candidates only (like in beam search). They sacrifice diversity for rationality by tightening the filtering parameter to truncate the undesirable “tail”

of the distribution, during which process an intersection point with metrics closest to human text is achieved (see the monotonicity in Figure 6 and 8, [Holtzman et al., 2020](#)). Clearly, diversity is always decreased in such scenario. Nor do they address the repetition issue that is caused by the “head” part of the distribution as is shown by our analysis, and if anything, even contribute to repetition, since after regularization on the filtered vocabulary, the originally high probability candidates are practically “amplified”. And most importantly, human text does not always favor high probability candidates. Inspired by these, we propose a heuristic method that enhances diversity in a different and “cooperative” way, that is, for the “head” part of the distribution, using inverse probability weighting to permute the distribution in order to suppress the “unnecessarily high” probability candidates that might be tedious and repetitive, and for the “tail” part, using multiple and dynamic filtering metrics to truncate unreliable tails in order to guarantee that the “head” part is correctly identified.

3 Interquartile Range Inverse Probability Sampling Algorithm

3.1 Subset Division on Filtered Vocabulary

In order to determine the boundary that indicates where are the “unnecessarily high” probability candidates (i.e., “head”) that we need to permute, we propose to adopt the commonly used interquartile range (IQR) method for the predicted distribution on the filtered vocabulary to identify “outlier” candidates. First, we need to ensure that only the most reliable candidates are kept in order not to interfere with the identification of “head” and later sampling. Following the common filtering method of stochastic sampling, we propose to jointly filter an initial subset V^{K_0} of candidates with p and k as follows.

$$V^{K_0} = V^k \cap V^p. \quad (3)$$

Let $p_{fil}(x)$ denote the regularized distribution on V^{K_0} . We propose to calculate IQR of $p_{fil}(x)$, that is, calculate 75% percentile as Q_3 , 25% percentile as Q_1 , $IQR = Q_3 - Q_1$, and divide V^{K_0} into subsets as follows.

IQR Subset Division of V^{K_0} :

$$\begin{aligned}
V^{VeryHigh} &: p_{fil}(x) \geq Q_3 + \rho \times IQR \\
V^{High} &: Q_3 + \rho \times IQR > p_{fil}(x) \geq Q_3 \\
V^{Medium} &: Q_3 > p_{fil}(x) \geq Q_1 \\
V^{Low} &: Q_1 > p_{fil}(x) \geq Q_1 - \rho \times IQR
\end{aligned} \tag{4}$$

where ρ is the hyper parameter for the coefficient of IQR with typical value being 1.5. Considering the characteristic of IQR, $V^{VeryHigh}$ can be regarded as the “head” part that we need to permute, which we expect that the least probable candidate in $V^{VeryHigh}$ is still likely to be “high enough” to be reasonable choices. Since IQR is based on quantile, $V^{VeryHigh}$ is mostly likely to be non-singleton on flat distribution, hence permutation on $V^{VeryHigh}$ will not interfere with peaked distribution which may compromise rationality of the distribution. See Appendix B for more discussions.

3.2 Deeper Look for “Tail” on Peaked Distribution

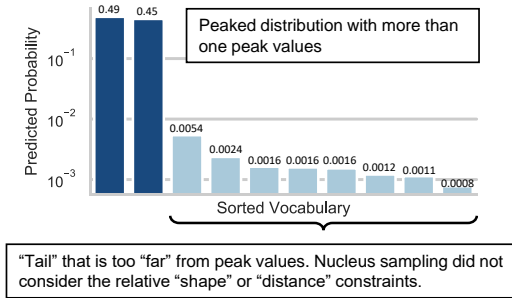


Figure 4: “Tail” part of peaked distribution that has more than one peak value. This distribution is also selected from one of the generated samples using GPT-2 Small model. On this distribution, small value of p for nucleus sampling will miss the second peak, while large value of p will easily let in low probability candidates below the peak.

Before proceeding, we take a deeper look for “tail” part on peaked distribution. As is studied by Holtzman et al. (2020), nucleus sampling can adaptively truncate low probability “tails” on peaked distribution, while top- k sampling can’t (see Figure 5, Holtzman et al., 2020). We consider a special case of peaked distribution in Figure 4 with more than one peak value, which is not considered by Holtzman et al. (2020). In this case, small value of p for nucleus sampling will miss the second peak, while large value of p will easily let in low proba-

bility candidates. This is because neither top- k sampling nor nucleus sampling considers the relative “shape” or “distance” constraints during filtering. To fix this, we propose a new filtering metric to further exclude low probability candidates that is too “far” from the peaked ones. We define a threshold that is the fraction of the maximum probability on a predicted distribution, and exclude candidates with probability below that threshold, which we name as the “top-1 controlled” (top1ctrl) filtering metric with parameter n as follows.

$$V^n = \{x \mid p(x) \geq \frac{\max p(x)}{n}, x \in V\}. \tag{5}$$

Clearly it is insufficient to use top1ctrl metric alone for vocabulary filtering, since directly applying small value of n on a flat distribution will also be problematic, and the purpose for this metric is to exclude very low probability candidates that happen on distributions like Figure 4 which is difficult to deal with using loose value of p or k . Consequently, we propose to use top1ctrl to prune V^{K_0} (on the basis on joint vocabulary filtering in Equation 3) in a dynamic way. Our method is described in the following equations, in which we denote the pruned set to be V^{K_1} .

$$V^{K_1} = \begin{cases} V^{VeryHigh} \cup V^{High}, & \text{if } V^n \subseteq (V^{VeryHigh} \cup V^{High}) \\ V^{K_0} \cap V^n, & \text{otherwise} \end{cases} \tag{6}$$

The first sub-equation ensures that V^n does not truncate any candidates categorized as “Very High” or “High”, since they are identified by IQR and likely to contain rational candidates. In this case we drop all candidates in V^{Medium} and V^{Low} , because they are considered too “far” from maximum value in the distribution. And the second sub-equation describes other cases where V^n works jointly with V^k and V^p in a straight-forward way. Practically n is set to a fairly loose value of 100 in our experiment in order to function correctly with top- k filtering and nucleus filtering and not to over-prune V^{K_0} .

3.3 Inverse Probability Permutation

With V^{K_1} acquired, we propose to re-assign probability mass for each candidate in $V^{VeryHigh}$ proportionally to its inverse probability, while keeping the sum of probability mass in $V^{VeryHigh}$ constant. In this way, distribution on $V^{VeryHigh}$ is rescaled

and has inverse monotonicity, while distribution on V^{K_1} still maintains the probability distribution feature. For simplicity, now let $p_{fil}(x)$ denote the regularized distribution on V^{K_1} . The transformation on $V^{VeryHigh}$ is described as follows.

$$p_{inv}(x) = \left(\sum_{x \in V^{VeryHigh}} p_{fil}(x) \right) \times \frac{p_{fil}(x)^{-1}}{\sum_{x \in V^{VeryHigh}} p_{fil}(x)^{-1}}, \quad (7)$$

where $p_{inv}(x)$ denotes the permuted distribution, and $p_{inv}(x)$ outside $V^{VeryHigh}$ remains the same as $p_{fil}(x)$. Finally the stochastic sampling is performed according to $p_{inv}(x)$. We refer to the above algorithm as the interquartile range inverse probability (IQR-IP) sampling algorithm. We summarize the main differences of our algorithm as follows.

- We use dynamic vocabulary filtering with 3 parameters (p , k , and n). This aims at guaranteeing the rationality of $V^{VeryHigh}$ and keeping the most reliable candidates in V^{K_1} .
- Distribution on $V^{VeryHigh}$ identified by IQR is permuted using Equation 7. This aims at improving diversity by decreasing the probability of tedious and possibly repetitive candidates with high probability and increasing the probability of reasonable but more surprising candidates with low probability in $V^{VeryHigh}$.

3.4 Total Variance Analysis

We provide total variance analysis to explain our algorithm. Following proposition by [Kang and Hashimoto \(2020\)](#), we can assess the permutation by analyzing the upper bound of total variance between $p_{inv}(x)$ and reference distribution $p_{ref}(x)$ with the following corollary.

Corollary 1. Upper bound of total variance between p_{inv} and p_{ref} satisfies

$$|p_{inv} - p_{ref}|^2 \leq \frac{1}{2} KL(p_{ref} || p_{fil}) + 2m + m^2, \quad (8)$$

where

$$m = \max_{x \in V^{VeryHigh}} \left| p_{fil} - \frac{Z_p}{p_{fil}} \right|, \quad (9)$$

$$Z_p = \frac{\sum_{x \in V^{VeryHigh}} p_{fil}}{\sum_{x \in V^{VeryHigh}} p_{fil}^{-1}}. \quad (10)$$

See Appendix A for proof.

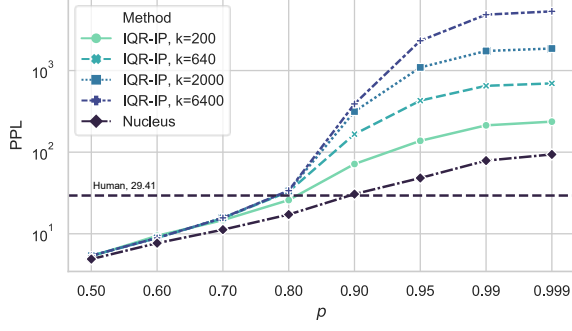
Equation 8 reveals an additional term controlled by m besides the original bound $\frac{1}{2} KL(p_{ref} || p_{fil})$ (achieved by p_{fil} without inverse probability transformation). Since m contains an value of inverse probability, the new upper bound will change dramatically. We argue that this provides a controllable diversity enhancement measure. Note that since $0 < Z_p \leq 1$, $\max |p_{fil} - \frac{Z_p}{p_{fil}}|$ can only be achieved on the largest or smallest value of p_{fil} in $V^{VeryHigh}$, i.e., on the first or last candidate of $V^{VeryHigh}$. As a result, m is controlled by ρ in Equation 4 and filtering parameters in Equation 6. For example, with a loosely filtered V^{K_1} , $V^{VeryHigh}$ might contain a last candidate with too small value of probability and render too large value of m , hence the total variance will become too high and corrupt the algorithm. However, with carefully chosen parameters, m may provide reasonable variation that enhances diversity and reduces repetition. We show this in the evaluation results.

4 Evaluation

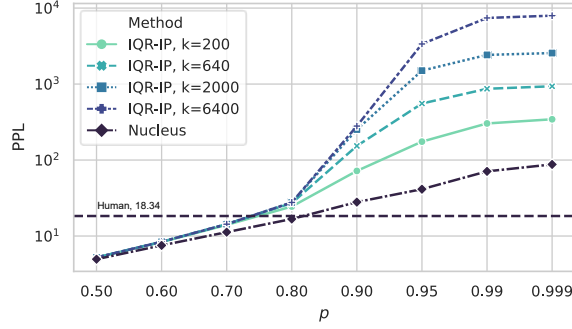
We use pre-trained GPT-2 Small (117M parameters) and GPT-2 XL (1,542M parameters) released by [Wolf et al. \(2019\)](#) to evaluate our algorithm. The primary goal is to compare our algorithm with nucleus sampling regarding diversity. We set fixed value of $n = 100$ for top1ctrl filtering and $\rho = 1.5$ for IQR. Following corresponding settings by [Holtzman et al. \(2020\)](#), we set maximum length of generation to be 200 and generate 5000 samples for each sampling method. We provide detailed automatic metrics to reveal the diversity gain property of our algorithm. A closer metric between the generated samples and human text indicates a greater resemblance. Generated examples are presented in Appendix B. We did not provide HUSE ([Hashimoto et al., 2019](#)) evaluation because we have 40 sets of sampling methods that will be intractable and expensive for human annotation.

4.1 Evaluation on Perplexity

As is shown in Figure 5, the perplexity of generated samples using our algorithm exhibits different tendency compared with nucleus sampling. They both achieve human level perplexity but with different filtering metric. This can be interpreted in several ways. First, our algorithm achieves human metric “earlier” with smaller p , i.e., with more strictly filtered vocabulary. Second, for the same value of p ,



(a) Perplexity for GPT-2 Small. Horizontal line (29.41, Radford et al., 2019) refers to human text.



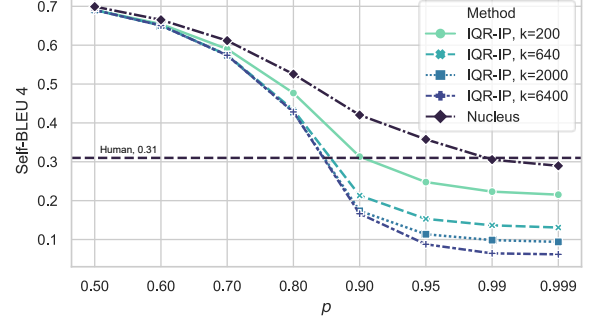
(b) Perplexity for GPT-2 XL. Horizontal line (18.34, Radford et al., 2019) refers to human text.

Figure 5: Comparison for perplexity.

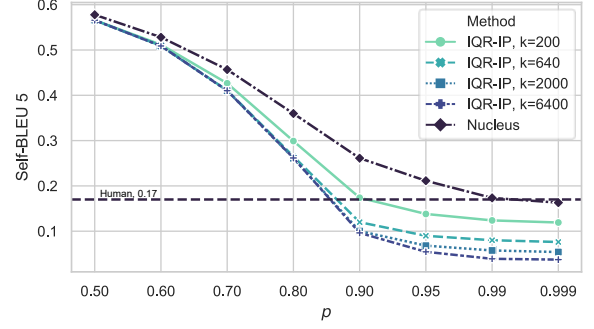
our algorithm achieves higher perplexity than nucleus sampling. And finally, the perplexity of our algorithm rises significantly faster. This is because the fast increase of additional term m from Corollary 1 when loosening the filtering. These results all indicate diversity gain compared with nucleus sampling. Note that such diversity gain will be destructive when $p > 0.9$, because the inverse value in term m will grow too big and “blow up” the algorithm. Thus the intersection points with human text is the reasonable choices for our algorithm.

4.2 Evaluation on Self-BLEU

Following the results by Holtzman et al. (2020) and Zhu et al. (2018), we compare the self-BLEU 4 and self-BLEU 5 scores where 1 in 5000 samples is calculated against all other samples for each sampling method. Lower score indicates higher diversity. As is shown in Figure 6 and 7, the self-BLEU scores achieved by our algorithm decrease significantly faster than nucleus sampling. Note that it can achieve almost the same score with “pure sampling” near $p = 0.999$ that represents highest diversity with traditional methods. Similar to results for perplexity, our algorithm achieves lower self-BLEU scores with smaller value of p , which

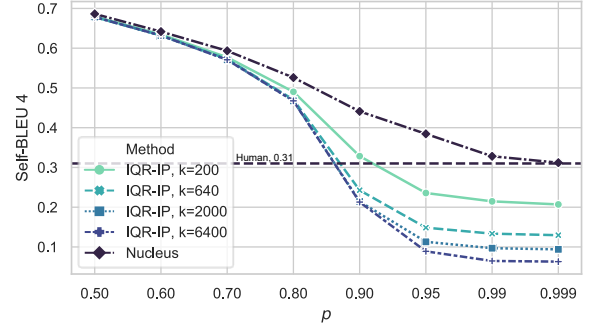


(a) Self-BLEU 4 for GPT-2 Small. Horizontal line (0.31, Holtzman et al., 2020) refers to human text.

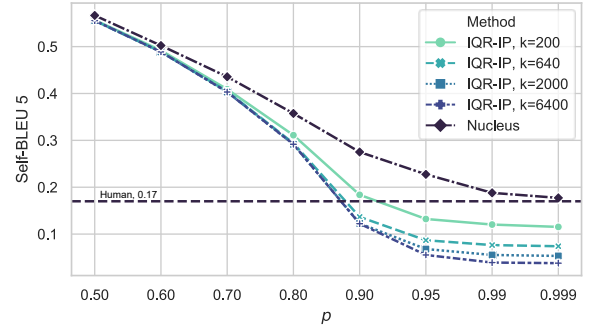


(b) Self-BLEU 5 for GPT-2 Small. Horizontal line (0.17, Holtzman et al., 2020) refers to human text.

Figure 6: Comparison for self-BLEU of GPT-2 Small



(a) Self-BLEU 4 for GPT-2 XL.

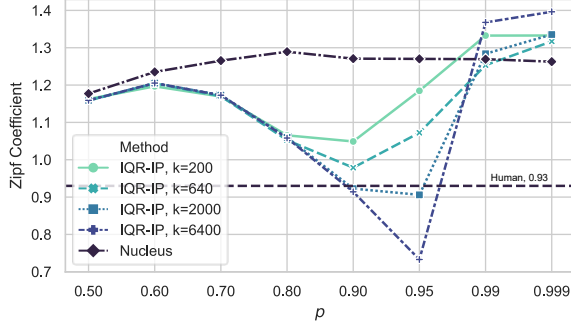


(b) Self-BLEU 5 for GPT-2 XL.

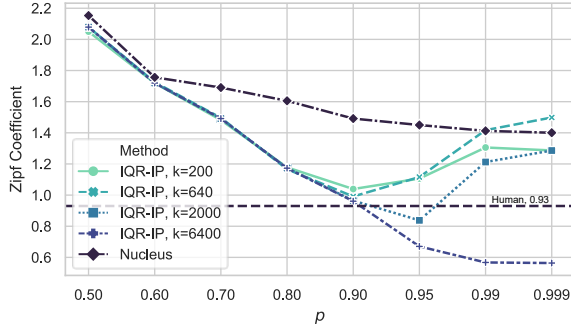
Figure 7: Comparison for self-BLEU of GPT-2 XL

also indicates diversity gain.

4.3 Evaluation on Zipf Coefficient



(a) Zipf Coefficient for GPT-2 Small.



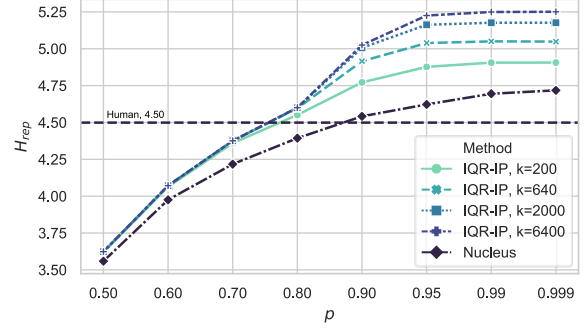
(b) Zipf Coefficient for GPT-2 XL.

Figure 8: Comparison for Zipf coefficient. Horizontal line (0.93, [Holtzman et al., 2020](#)) refers to human text.

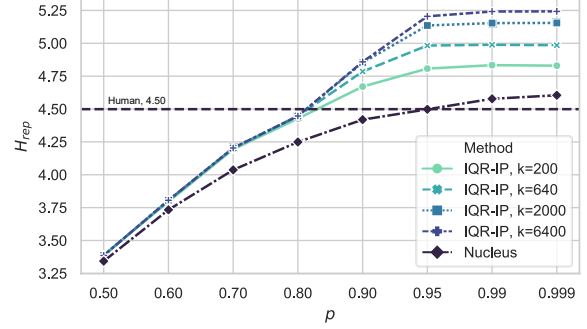
We also follow works by [Holtzman et al. \(2020\)](#) and [Piantadosi \(2014\)](#) to evaluate Zipf coefficient which represents linguistic feature for word frequency distribution, where Zipf’s Law takes the following form.

$$f(w) \propto \frac{1}{(r + \beta)^\alpha}, \quad (11)$$

where r denotes the frequency rank of word w , $f(w)$ denotes the word frequency in a give corpus, and α is referred to as the Zipf coefficient. We fit α on samples generated by different sampling methods. As is shown in Figure 8, our algorithm can fit almost identical α to human text, while nucleus sampling can’t. It also shows that our algorithm has inflection point for Zipf coefficient, unlike flat curve for nucleus sampling. This can also be interpreted that large value of p corrupts the algorithm and deviates from human text, and suitable p around the inflection point achieves closest resemblance to human text. This indicates that the permutation of our algorithm provides a very different distribution of vocabulary that is unable to achieve with nucleus sampling.



(a) H_{rep} for GPT-2 Small.



(b) H_{rep} for GPT-2 XL.

Figure 9: Comparison for H_{rep} . Horizontal line (4.50) refers to human text on test set of WikiText-2

4.4 Evaluation on Repetition

Different from [Holtzman et al. \(2020\)](#), we take an explicit metric to evaluate repetition. We calculate the entropy of frequency distribution of tokens in a fixed-length token window as follows.

$$H_{rep} = - \sum_w p(w) \times \log p(w), \quad (12)$$

where

$$p(w) = \frac{f(w)}{\sum_w f(w)}, \quad (13)$$

where $f(w)$ denotes the frequency of word w within a 200-length token window. This metric will represent the repetition tendency as well as the word frequency distribution feature. For example, samples with repetition loops will have concentrated distribution of $p(w)$ hence having lower H_{rep} , while samples with diverse usage of vocabulary will have flat distribution of $p(w)$ hence having higher H_{rep} . We use test set of WikiText-2 to calculate metrics for human text, and use sliding window with stride being 200 to calculate the average on all windows. Results in Figure 9 are similar to results for perplexity, which show that H_{rep} of our algorithm grows faster and stays higher than nucleus

sampling, which represents more diverse usage of vocabulary and less repetition.

5 Conclusion

In this work we propose the interquartile range inverse probability sampling algorithm. It brings reasonable permutation on the predicted distribution of filtered vocabulary to enhance diversity. We evaluate our algorithm with pre-trained language models. Results show that our algorithm can generate samples with higher diversity and less repetition compared with traditional methods.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#).
- Imre Csizsár and János Körner. 2011. *Information Theory: Coding Theorems for Discrete Memoryless Systems*, 2 edition. Cambridge University Press.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. [Transformer-XL: Attentive language models beyond a fixed-length context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- T. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *ArXiv*, abs/1904.02792.
- Hernán MA, Robins JM. 2020. *Causal Inference: What If*. Chapman & Hall/CRC, Boca Raton.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. [Learning to write with cooperative discriminators](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.
- Daniel Kang and Tatsunori Hashimoto. 2020. [Improved natural language generation via loss truncation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. [Deep reinforcement learning for dialogue generation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.
- S.T. Piantadosi. 2014. [Zipf’s word frequency law in natural language: A critical review and future directions](#). *Psychonomic Bulletin & Review*, 21:1112–1130.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. [Style transfer from non-parallel text by cross-alignment](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. [Tegygen: A benchmarking platform for text generation models](#). In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’18*, pages 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Proof of Corollary

First, with Pinsker’s inequality (Csiszár and Körner, 2011), the total variance between the original filtered distribution p_{fil} and the reference distribution p_{ref} satisfies

$$|p_{fil} - p_{ref}|^2 \leq \frac{1}{2} KL(p_{ref} || p_{fil}). \quad (14)$$

Then we may use similar methods by Kang and Hashimoto (2020) to derive the new bound as follows.

Proof.

$$|p_{inv} - p_{ref}|^2 \leq (|p_{inv} - p_{fil}| + |p_{fil} - p_{ref}|)^2 \quad (15)$$

By definition of p_{inv} in Equation 7, we have

$$|p_{inv} - p_{fil}|^2 \leq \max_{x \in V^{VeryHigh}} |p_{fil} - \frac{Z_p}{p_{fil}}|. \quad (16)$$

Then expand Equation 15, and use m defined in Equation 9 and 16 to bound $|p_{inv} - p_{fil}|$, and use Equation 14 to bound $|p_{fil} - p_{ref}|$, the inequality is proved. \square

This corollary has the same form as Kang and Hashimoto (2020), although with different constant m , which corresponds to the truncation ratio c of their proposition. In our work, m is controlled by inverse probability transformation and can be fairly large, while the truncation ratio c satisfies $0 \leq c \leq 1$. In this way, it can be regarded as an extension from proposition by Kang and Hashimoto (2020) in a different scenario.

B Generated Examples and Further Discussion

We present several samples generated by GPT-2 Small and GPT-2 XL selected from our experiment in Table 1. For fair comparison, we choose sampling strategies that are near the intersection point for human text in the perplexity evaluation from Figure 5. It could be seen that our algorithm favors creating obscure and surprising sentences more than traditional methods. Such difference of language style is caused by the inverse probability transformation from Equation 7.

A possible concern of IQR is whether it will interfere with peaked distribution that has only a few reasonable candidates (e.g., 1 or 2) with high probability in V^{K_0} . Note that by definition of IQR, it will only put “outliers” in $V^{VeryHigh}$. Clearly,

for V^{K_0} with less than 4 candidates, they will be partitioned among the “middle part” of subsets, i.e., symmetrically distributed on V^{High} , V^{Medium} and V^{Low} . As a result, on highly peaked distribution with only a few “unquestionably correct” candidates with high probability in V^{K_0} , there will be no $V^{VeryHigh}$ as we have observed, which means that the inverse probability part won’t work and the algorithm will degrade into plain stochastic sampling. This indicates that IQR can adaptively work on peaked distribution without compromising fluency. Another issue to clarify is that by the definition of IQR, there should be a $V^{VeryLow}$ that locates symmetrically to $V^{VeryHigh}$ on the identification range. In our experiment we found that this boundary is always below 0, i.e., $V^{VeryLow}$ is always empty set during IQR calculation. As a result, we omit the narration for $V^{VeryLow}$.

Also note that one may observe some undesirable obscureness in generated samples. To tune this, note that ρ in Equation 4 can be used to control the identification range for $V^{VeryHigh}$, hence controlling the intensity for inverse probability transformation, i.e., controlling the diversity gain that results in style difference. For example, one may need to tune ρ from 1.5 up to 2.0 or even higher, if the generated texts seem to lose fluency and have too many obscure sentences (this may be highly likely to happen with other generation model, since the parameters we provide only suit for GPT-2). If ρ is set to infinity, there will be no $V^{VeryHigh}$ and our algorithm will degrade to plain stochastic sampling filtered by Equation 3 and 6. Practically speaking, with a fixed filtering method that determines the filtered vocabulary to sample on (no matter with top- k sampling only, nucleus sampling only, or multiply constraints like ours in Equation 3 and 6), applying inverse probability transformation using Equation 4 and 7 with lower ρ (e.g., 1.5 or lower) will enlarge $V^{VeryHigh}$ hence tending to create more surprising and obscure sentences away from traditional methods (this may be more suitable for artistic generation such as poetry or music generation), and higher ρ (e.g., 2.0 or higher) will shrink $V^{VeryHigh}$ hence tending to create more plain and common sentences close to traditional methods (this may be more suitable for tasks with strict context constraints such as summarization or translation).

As a matter of fact, one may even design different and more “mild” permutation strategies besides Equation 7, e.g., evenly redistributing $V^{VeryHigh}$,

Human	<p>She walks in beauty, like the night <\n> Of cloudless climes and starry skies; <\n> And all that's best of dark and bright <\n> Meet in her aspect and her eyes; <\n> Thus mellowed to that tender light <\n> Which heaven to gaudy day denies. <\n> One shade the more, one ray the less, <\n> Had half impaired the nameless grace <\n> Which waves in every raven tress, <\n> Or softly lightens o'er her face; <\n> Where thoughts serenely sweet express, <\n> How pure, how dear their dwelling-place. <\n> And on that cheek, and o'er that brow, <\n> So soft, so calm, yet eloquent, <\n> The smiles that win, the tints that glow, <\n> But tell of days in goodness spent, <\n> A mind at peace with all below, <\n> A heart whose love is innocent! <\n></p>
GPT-2 XL, IQR-IP $p=0.7$, $k=200$ <i>(ours)</i>	<p>She walks in beauty, <\n> like a virgin on a snow-white carpet, <\n> with an elegant train, <\n> a very elegant train, <\n> like the snow-white, white marble <\n> of a private train. <\n> A fine-looking man with an air of superiority, <\n> and one who looks a great deal like the Empress Dowager Cixi, <\n> dressed in a long, dark gown, <\n> on his shoulders, <\n> the long, dark cloak, <\n> flowing with an amorous light, <\n> leaving the shadow of a cloud, <\n> and leaving behind the air of mystery, <\n> and leaving behind the veil of beauty. <\n> Now this gentleman comes, and his person is also <\n> like the Emperor, who has been given <\n> the Imperial dress of silver and gold. <\n> The dress of silver and gold, <\n> as it flows over the man, <\n> like a waterfall,</p>
GPT-2 XL, nucleus $p=0.8$	<p>She walks in beauty to those in mourning. She walks among the dead with an undying grace. <u>She walks among the dead with an undying grace. She walks among the dead with an undying grace. She walks among the dead with an undying grace. She walks among the dead with an undying grace.</u> (...)</p>
GPT-2 Small, IQR-IP $p=0.8$, $k=200$ <i>(ours)</i>	<p>She walks in beauty at once. A simple life could not exist without me. My words could not possibly exist without you. She watches as my thoughts, even if their sound and pain and grief echo across the ocean and sky, build on the last gasp of hope that gave life to this island. Her life could not take a stand against us. It is something we will always hold on to. She will be right back where she left off. It will be the best thing for the family to do for her. Her arms hug my chest. Her mother. My father. We walk in the bright sunshine, laughing and praying. I love the sunshine. It's what I always do. As she leaves, she makes her way back into the ship. My life can never change. I am lost.</p>
GPT-2 Small, nucleus $p=0.9$	<p>She walks in beauty so beautifully, loved by our dear Abby, that only we can truly imagine how much she touched them. And she goes where she needs to go in places that we never know—she cuts lines! And we can always tell we are not alone, and will never be alone in our love.</p>

Table 1: Generated examples using different sampling methods that have average perplexity near human text in Figure 5. Repetition is marked in red with underline. While nucleus sampling generates plain and occasional repetitive sentences using more loosely filtered vocabulary and lots of unsurprising high-frequency words, our algorithm generates more diverse, more surprising and even obscure sentences with more strictly filtered vocabulary.

or simply adding some noise on $V^{VeryHigh}$, to achieve a less severe permutation bounded by Equation 16. In this way, our algorithm is actually an extreme case that we completely re-order $V^{VeryHigh}$ with inverse probability and try to bring significant permutation on the filtered vocabulary.