Improving Diversity of Neural Text Generation via Inverse Probability Weighting

Xinran Zhang\textsuperscript{1}, Maosong Sun\textsuperscript{12,*}, Jiafeng Liu\textsuperscript{1}, Xiaobing Li\textsuperscript{1}
\textsuperscript{1}Department of Music Artificial Intelligence and Music Information Technology
Central Conservatory of Music, Beijing, China
\textsuperscript{2}Department 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 text generation suffers from the text degeneration issue such as repetition. Traditional stochastic sampling methods only focus on truncating the unreliable “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 consider the issue that human text does not always favor high-probability words. Inspired by these, in this work we propose a heuristic sampling method. We propose to use interquartile range of the predicted distribution to determine the “head” part, then permutate and rescale the “head” with inverse probability. This aims at decreasing the probability for the tedious and possibly repetitive candidates with high probability, and increasing the probability for the rational but more surprising candidates with lower probability. The proposed algorithm provides a reasonable permutation on the predicted distribution which enhances diversity without compromising the rationality of the distribution as well as the fluency of the generated text. Experiment results show that our algorithm can increase diversity while achieving close resemblance to human text compared with traditional methods.

1 Introduction

Neural text generation is an important natural language processing (NLP) task, and have benefited a lot from Transformer (Vaswani et al., 2017) architecture. However, it suffers from the well-known text degeneration issue (Holtzman et al., 2020), that is, the decoded texts exhibit a strong tendency to be repetitive with low diversity. To address this, many works have focused on stochastic sampling by truncating the “tail” of the distribution, e.g., the top-k sampling (Fan et al., 2018; Holtzman et al., 2020) or nucleus sampling (top-p sampling, Holtzman et al., 2020), which directly truncates the predicted distribution during sampling, excluding unreliable “tail” with low probability. Recent work by Basu et al. (2021) adaptively truncates the “tail” to achieve controllable quality. Regrettably, none of these methods have directly addressed the discrepancy 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 in our analysis that repetitive samples with low diversity are actually caused by the “head” part with high probability. Inspired by this, 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 without compromising the rationality of the distribution as well as the fluency of the generated text. Experiment results show that our algorithm can increase diversity while achieving close resemblance to human text compared with traditional methods.

2 Observation on the “Tail” and “Head”

2.1 Traditional Methods: Truncating the “Tail” to Balance between Quality and Diversity

As is widely acknowledged for text generation, directly sampling on the predicted distribution will produce unsatisfactory samples due to the low-probability “tail” of the distribution (Holtzman et al., 2020). This is self-explanatory since the “tail” contains unreasonable words that lead to less repetition as well as lower quality. Consequently, traditional methods always start by truncating the “tail”. For example, the top-k sampling (Fan et al., 2018; Holtzman et al., 2018) filters the top k probable candidates from the vocabulary (denoted by V) as follows.

\[ V^k = \{ x \mid \text{rank}(p(x)) \leq k, x \in V \} \tag{1} \]
where \( p(x) \) denotes the predicted distribution of the language model, and \( \text{rank} \) 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.

According to Holtzman et al. (2020), top-\( k \) sampling cannot address the discrepancy between peaked distribution and flat distribution. They propose nucleus sampling (top-\( p \) sampling) which filters the vocabulary with top \( p \) mass of cumulative probability as follows.

\[
V^p = \{ x \mid \text{cdf}(x) \leq p, x \in V \}, \tag{2}
\]

where the cumulative density function \( \text{cdf}(x) \) is calculated on the sorted distribution of \( p(x) \). This produces better results than top-\( k \) sampling, because it can dynamically drop more “tails” on peaked distribution, while top-\( k \) sampling can’t.

Clearly, these methods balance between quality and diversity by truncating the “tail”. Dropping more “tails” dynamically like nucleus sampling will improve quality but result in more repetition and lower diversity (see Table 1, Holtzman et al., 2020), while keeping more “tails” like top-\( k \) sampling will achieve less repetition but lower quality. Recent methods such as MIROSTAT by Basu et al. (2021) adaptively truncate the “tail” with pre-defined quality target (perplexity) for better balancing effect.

### 2.2 Repetition Loops Caused by the “Head”

However, traditional methods do not address the “head” part, which we show may lead to the annoying repetition loops.

To explore the behavior of repetition loops, we use GPT-2 Small (Radford et al., 2019) with nucleus sampling (\( p = 0.95 \)) to generate 5,000 samples with the same input context and set maximum generation length to be 1,024. The sharing input context is “She walks in beauty” (from Lord Byron’s most famous poetry).

To detect repetition as well as measuring the concentration tendency of vocabulary, we use a very straightforward metric by calculating the entropy of word distribution in a fixed-length window as follows.

\[
H_{\text{rep}} = - \sum_w p(w) \times \log p(w), \tag{3}
\]

\[
p(w) = f(w)/\sum_w f(w), \tag{4}
\]

where \( f(w) \) denotes the frequency of word \( w \). Samples with repetition loops will have concentrated distribution of \( p(w) \) hence having lower \( H_{\text{rep}} \), while samples with diverse usage of vocabulary will have flat distribution of \( p(w) \) hence having higher \( H_{\text{rep}} \). Empirically, we use \( H_{\text{rep}} < 2 \) for all 200-length token windows to detect repetitive passages for observation.

We present a very representative sample that contains infinite loops of “She walks in beauty” (with generated period). The trajectory of first 3 generated loops is presented in Figure 1. We found several phenomena that cause this repetition.

- Repetitive candidates always have high probability and high rank in the predicted distribution (see “*” labeled candidates in each heatmap box in Figure 1).
- 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 distribution 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.
- The predicted distribution got stuck in extreme distribution that assigns almost all probability mass for repetitive candidates (e.g., “in” and “beauty” in Loop #3).

To further verify these phenomena, we extract and align the trajectories of each repetitive words to observe the overall trajectory for repetitive words (e.g., aligning all appearances of “She” sequentially on the \( x \) axis). 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 distribution where predicted probability approaches 1, rank approaches 1, and entropy 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
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 the first 3 repetition loops. 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 token windows. 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 the model to extreme distribution that causes repetition loops.

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.

Clearly, these undesirable behaviors of the “head” with high probability will lead the model to generate samples that might contain repetition loops with low diversity. Regrettably, this issue is unable to address by tradition stochastic sampling algorithms, since they still encourage to sample on high-probability candidates.

2.3 Improving Diversity by Permutating the “Head” on Flat Distributions

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). Our results in Section 2.2 also show that it will be harmful to sample according to likelihood of candidates due to the behavior of the “head”.

To fix this, we present a detailed observation of the “head” in Figure 3. It shows that lower-probability candidates on a flat distribution are actually reasonable but more surprising with higher diversity. Consequently, it is possible to increase diversity by emphasizing on less probable candidates on flat distributions without compromising the rationality of the distribution as well as the fluency of the generated text.

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 (i.e., the “head”) 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 without
IQR Subset Division of $V^{K_0}$:

\[
\begin{align*}
V^{V_{VeryHigh}} & : p_{fil}(x) \geq Q_3 + \rho \times IQR \\
V^{V_{High}} & : Q_3 + \rho \times IQR > p_{fil}(x) \geq Q_3 \\
V^{V_{Medium}} & : Q_3 > p_{fil}(x) \geq Q_1 \\
V^{V_{Low}} & : Q_1 > p_{fil}(x) \geq 0.1
\end{align*}
\]

where \(\rho\) is the hyper parameter for the coefficient of IQR with typical value being 1.5. Considering the outlier-identification nature of IQR, \(V^{V_{VeryHigh}}\) can be regarded as the “head” part that we need to permute, which we expect that the least probable candidate in \(V^{V_{VeryHigh}}\) is still likely to be “high enough” to be reasonable choices.

Since IQR is based on quantile, \(V^{V_{VeryHigh}}\) is empirically to be non-singleton on flat distribution only, hence permutation on \(V^{V_{VeryHigh}}\) will not interfere with peaked distribution which may compromise rationality of the distribution. See Appendix C for more discussions.

3.2 “Leakage” of the “Tail” on Peaked Distribution Interferes with Identification of the “Head”

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 which is not considered by Holtzman et al. (2020). Figure 4 presents an actual example of peaked distribution with more
than one peak value. 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-probability candidates, i.e., resulting in “leakage”. Although such leakage might affect very little on sampling (since the “leakage” part has low probability), but clearly it will affect the identification of “head” (since IQR calculation is based on quantile), hence cannot be ignored.

We argue that the incurring of leakage 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.

\[
V^n = \{ x \mid p(x) \geq \max p(x)/n, x \in V \}. \quad (7)
\]

We propose to use this metric to prune \( V^{K_0} \) (on the basis on joint vocabulary filtering in Equation 5) in a dynamic way. Our method is described in the following equations, in which we denote the pruned set to be \( V^{K_1} \).

\[
V^{K_1} = \begin{cases} 
V^{VeryHigh} \cup V^{High}, & \text{if } V^n \subseteq V^{VeryHigh} \cup V^{High} \\
V^{K_0} \cap V^n, & \text{otherwise}
\end{cases} \quad (8)
\]

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 on the “Head”

With \( V^{K_1} \) acquired, we propose to re-assign probability mass for each candidate in \( V^{VeryHigh} \) (i.e., the “head”) proportionally to its inverse probability, while keeping the sum of probability mass in \( V^{VeryHigh} \) constant. In this way, distribution of the “head” 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 permutation on \( V^{VeryHigh} \) is described as follows.

\[
p_{inv}(x) = \left( \sum_{x \in V^{VeryHigh}} p_{fil}(x) \right) \times \frac{p_{fil}(x)^{-1}}{\sum_{x \in V^{VeryHigh}} p_{fil}(x)^{-1}}, \quad (9)
\]

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 correct identification of the “head” of the distribution.
- Distribution of the “head” identified by IQR is permuted using Equation 9. 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.

3.4 Total Variance Analysis

We provide total variance analysis to explain the behavior of our algorithm. Following proposition by Kang and Hashimoto (2020), we can evaluate 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

\[
|p_{inv} - p_{ref}|^2 \leq \frac{1}{2} KL(p_{ref}||p_{fil}) + 2m + m^2, \quad (10)
\]

where

\[
m = \max_{x \in V^{VeryHigh}} |p_{fil} - \frac{Z_p}{p_{fil}}|, \quad (11)
\]

\[
Z_p = \frac{\sum_{x \in V^{VeryHigh}} p_{fil}}{\sum_{x \in V^{VeryHigh}} p_{fil}^{-1}}. \quad (12)
\]
See Appendix A for proof.

Equation 10 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 permutation). Since \( m \) contains an value of inverse probability, the new upper bound will change dramatically. This provides a controllable diversity enhancement measure. See Appendix B for more analysis.

4 Evaluation

4.1 Experiment Setup

The primary goal of the evaluation is to test whether our methods generate fluent samples with higher diversity. We consider the following principles when choosing baselines.

- **Ablation of permutating the “head”**. This means the baseline method should be without permutation, i.e., choosing plain stochastic sampling that only truncates the “tail” for comparison.

- **Fair comparison on human-level PPL**. This means the baseline method as well as our method should already achieve close PPL to human text like Figure 6 by Holtzman et al. (2020), i.e., choosing hyper parameters near the intersection points with human PPL for fair comparison.

- **Impact of model size**. This answers the question that does model size affect the conclusion from our experiments. We choose the smallest and largest plain auto-regressive Transformer language models from GPT-2 family for interpolative conclusions.

We use pre-trained GPT-2 Small (117M parameters) and GPT-2 XL (1,542M parameters) released by Wolf et al. (2019). Following identical settings by Holtzman et al. (2020), we set maximum length of generation to be 200 and generate 5,000 samples for each sampling method with the same context in Section 2.2. We set fixed value of \( n = 100 \) for \( \text{top1ctrl} \) filtering and \( \rho = 1.5 \) for IQR.

4.2 Statistical Evaluation

We first follow the statistical evaluation procedure by Holtzman et al. (2020), which evaluates the following metrics (closer score to the human metric is better).

- **Perplexity**. This metric is calculated on the generated texts with the per-trained model to reflect its general quality and fluency. Lower score indicates higher quality.

- **Self-BLEU (4 and 5)** (Holtzman et al., 2020; Zhu et al., 2018). One sample is calculated against all other samples to reflect diversity among all samples. Lower score indicates higher diversity.

- **Zipf coefficient** (Zipf, 1949; Newman, 2005). This metric represents linguistic feature of word frequency distribution. Lower score indicates more flat distribution of words and higher diversity.

- **Repetition**. We directly take \( H_{rep} \) from Equation 3 to evaluate repetition tendency, which reflects diversity within the sample. Higher score indicates less repetition and higher diversity.

As is shown in Figure 5(a) and 5(f), the PPL of generated samples using our algorithm can also achieve human level perplexity but with more strictly filtered vocabulary, which means our algorithm truncates more low-probability “tails” and still achieves equal PPL to human text, which is a desirable feature, since low-probability “tails” that contain unreasonable candidates will lower the quality of the generated text. This indicates that our algorithm relieves text degeneration not only by letting in the “tails” but also by permutating the “head”, unlike traditional methods that solely rely on the “tails”.

Note that our algorithm is highly sensitive to filtering metric, which is caused by the fast increase of additional term \( m \) from Corollary 1 when loosening the filtering. Such diversity gain will be destructive (e.g., for \( p > 0.9 \)), because the inverse value in term \( m \) will grow too big and “blow up” the algorithm. Thus the intersection points with human PPL is the reasonable choices for our algorithm.

As is clearly shown in Figure 5(b) and 5(h), the Self-BLEU scores achieved by our algorithm decrease significantly faster than nucleus sampling, which indicates great diversity gain. Note that it can achieve almost the same score with “pure sampling” near \( p = 0.999 \) that represents highest diversity in traditional methods. This means that our
Table 1: Statistical evaluation (closer metric to human text is better) and human evaluation (higher score is better) for selected decoding parameters. Note that our algorithm can achieve human level PPL with less repetition (with high $H_{rep}$). Also note the Zipf coefficient of our algorithm is much closer to human metric and unable to achieve by traditional methods. Human evaluation shows that our algorithm can achieve similar fluency but higher diversity.

Figure 5: Statistical results and metric behavior comparison with nucleus sampling. They show that our algorithm achieves human level metrics with more strict filtering parameters (i.e., with less “tail”), which is contributed by the diversity gain from inverse probability permutation on the “head”. Also note that the behavior of Zipf coefficient of our algorithm (with intersection to human metric) is significantly different from nucleus sampling (without intersection), because our algorithm encourages to sample on less probable tokens of the “head” which renders more flat distribution of the vocabulary and achieves closer resemblance to human text.

algorithm achieves significantly higher diversity but with less “tails”.

As is shown in Figure 5(d) and 5(i), our algorithm can fit identical Zipf coefficient to human text, while nucleus sampling can’t. This indicates that the permutation of our algorithm renders more flat vocabulary distribution (by encouraging sampling on less probable tokens) which is less concentrated, closer to human text and is unable to achieve by plain stochastic sampling (which always picks high-probability candidates and results in peaked and less diverse usage of vocabulary).

Results for repetition are shown in Figure 5(e) and 5(j). Similar to results for Self-BLEU scores, they also show that $H_{rep}$ of our algorithm grows faster and stays higher than nucleus sampling, which represents less repetition and higher diversity.

4.3 Human Evaluation

We collect 117 copies of human annotations per each sampling algorithm on fluency (focusing on...
She walks in beauty, like the night
\(\text{\textbullet} \) Of cloudless climes and starry skies;
\(\text{\textbullet} \) And all that’s best of dark and bright
\(\text{\textbullet} \) Meet in her aspect and her eyes.
\(\text{\textbullet} \) Thus mellowed to that tender light
\(\text{\textbullet} \) Which heaven to gaudy day denies.
\(\text{\textbullet} \) One shade the more, one ray less,
\(\text{\textbullet} \) \(\text{\textbullet} \) Had half impaired the nameless grace
\(\text{\textbullet} \) Which waves in every raven tress,
\(\text{\textbullet} \) \(\text{\textbullet} \) Or softly lightens o’er her face;
\(\text{\textbullet} \) \(\text{\textbullet} \) Where thoughts serenely sweet express,
\(\text{\textbullet} \) \(\text{\textbullet} \) How pure, how dear their dwelling
\(\text{\textbullet} \) \(\text{\textbullet} \) How calm, how clean, how cool!
\(\text{\textbullet} \) The smile that win, the tints that glow,
\(\text{\textbullet} \) \(\text{\textbullet} \) But tell of days in goodness spent,
\(\text{\textbullet} \) \(\text{\textbullet} \) A mind at peace with all below, \(\text{\textbullet} \) A heart whose love is innocent!

| Model | p | Human |
|-------|---|-------|
| GPT-2 Small, nucleus | 0.9 | She walks in beauty wheel chair (The Art Of Sleeping). Despite her excessive masculinity, who knows how she’ll dress. Whispered last night by Georgia. See full summary » |
| GPT-2 Small, Top-4, IQR-IP | 0.8 | She walks in beauty wheel chair (The Art Of Sleeping). Despite her excessive masculinity, who knows how she’ll dress. Whispered last night by Georgia. See full summary » |
| GPT-2 XL, nucleus | 0.9 | She walks in beauty and elegance, and she is a natural leader. Weiner’s future with the department is still undecided. The Post’s full story continues here. |
References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. MIROSTAT: A neural text decoding algorithm that directly controls perplexity. In International Conference on Learning Representations.

Imre Csiszár and János Körner. 2011. Information Theory: Coding Theorems for Discrete Memoryless Systems, 2 edition. Cambridge University Press.

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.

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 degeneration. 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.

Mark EJ Newman. 2005. Power laws, pareto distributions and zipf’s law. Contemporary physics, 46(5):323–351.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz 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.

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.

Yaoming Zhu, Sidi Lu, Lei Zheng, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: 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.

George K. Zipf. 1949. Human Behaviour and the Principle of Least Effort. Addison-Wesley.
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

$$|p_{fil} - p_{ref}|^2 \leq \frac{1}{2} KL(p_{ref} || p_{fil}). \quad (13)$$

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

Proof.

$$|p_{inv} - p_{ref}|^2 \leq (|p_{inv} - p_{fil}| + |p_{fil} - p_{ref}|)^2 \quad (14)$$

By definition of $p_{inv}$ in Equation 9, we have

$$|p_{inv} - p_{fil}|^2 \leq \max_{x \in \text{VeryHigh}} |p_{fil} - \frac{Z_p}{p_{fil}}|. \quad (15)$$

Then expand Equation 14, and use $m$ defined in Equation 11 and 15 to bound $|p_{inv} - p_{fil}|$, and use Equation 13 to bound $|p_{fil} - p_{ref}|$, the inequality is proved. \hfill \Box

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 permutation 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.

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 $V_{\text{VeryHigh}}$, i.e., on the first or last candidate of $V_{\text{VeryHigh}}$. As a result, $m$ is controlled by $\rho$ in Equation 6 and filtering parameters in Equation 8. For example, with a loosely filtered $V^K_i$, $V_{\text{VeryHigh}}$ 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, as is shown in the evaluation results.

B Ablation Study

We present ablation study of IQR coefficient and $top1ctrl$ filtering in Table 3. Clearly, when $\rho$ in Equation 6 increases, it shortens the identification range of $V_{\text{VeryHigh}}$ hence decreasing the intensity of inverse probability weighting, which leads to more repetition (with higher Self-Bleu score and lower $H_{\text{rep}}$), more concentrated distribution of vocabulary (with higher Zipf coefficient), more plain and unsurprising sentences (with lower PPL). As a result, $\rho$ can be used to control the diversity gain that results in style difference. For example, one may need to tune $\rho$ to higher values, if the generated texts seem to lose fluency and have too many obscure sentences (this may be more suitable for artistic generation that requires high diversity and creativity such as poetry or music generation). If $\rho$ is set to infinity, there will be no $V_{\text{VeryHigh}}$ and our algorithm will degrade to plain stochastic sampling filtered by Equation 5 and 8 (this may be more suitable for tasks that require high fluency such as summarization or translation).

For the ablation of $top1ctrl$ filtering, Table 3 clearly shows that loosening $n$ will be harmful, since generated samples will lose quality (with higher PPL). Although this results in less repetition and higher diversity (with lower Self-Bleu score and higher $H_{\text{rep}}$), but clearly due to the “leakage” of tail described in Section 3.2, the diversity gain will be destructive which is introduced by candidates with too low probability that interfere with the identification of $V_{\text{VeryHigh}}$, which is also reflected by the decrease of Zipf coefficient that represents more flat distribution of vocabulary. On the other hand, small value of $n$ will over-prune the vocabulary, which indirectly decreases the range of $V_{\text{VeryHigh}}$ hence decreasing the intensity of inverse probability weighting, resulting in lower

| Method      | PPL   | Self-BLEU | Self-BLEU | Zipf Coef. | $H_{\text{rep}}$ |
|-------------|-------|-----------|-----------|------------|-----------------|
| GPT-2 XL    |       |           |           |            |                 |
| IQR-IP      | 16.77 | 0.47      | 0.29      | 1.17       | 4.45            |
| $\rho = 0.8$ | $k = 640$ |           |           |            |                 |
| $n = 100$   |       |           |           |            |                 |
| $\rho = 1.5$ |       |           |           |            |                 |
| $\rho = 3.0$ |       |           |           |            |                 |
| $\rho = 5.0$ |       |           |           |            |                 |
| $\rho = 10.0$ |       |           |           |            |                 |
| $\rho = 50.0$ |       |           |           |            |                 |

Table 3: Ablation study of IQR coefficient and $top1ctrl$ filtering for GPT-2 XL.
diversity and more repetition.

C  Further Explanations on IQR

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 permutation won’t work and the algorithm will degrade into plain stochastic sampling. This indicates that IQR can adaptively work on flat distribution and 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 $V^{VeryLow}$.

Note that one may even design different and more “mild” permutation strategies besides Equation 9, e.g., evenly redistributing $V^{VeryHigh}$, or simply adding some noise on $V^{VeryHigh}$, to achieve a less severe permutation bounded by Equation 15. In that case, our algorithm is actually an extreme case that we completely re-order $V^{VeryHigh}$ with inverse probability which brings significant permutation on the predicted distribution.