Balancing Diversity and Risk in LLM Sampling: How to Select Your Method and Parameter for Open-Ended Text Generation

Anonymous ACL submission

Abstract

Sampling-based decoding strategies have been widely adopted for Large Language Models (LLMs) in numerous applications, targeting a balance between diversity and quality via temperature tuning and tail truncation. Considering the strong dependency of the candidate next tokens on different prefixes, recent studies propose to adaptively truncate the tail of LLMs' predicted distribution. Although improved results have been reported with these methods on open-ended text generation tasks, the results are highly dependent on the curated parameters and the limited exemplar text. In this paper, we propose a systematic way to estimate the capacity of a truncation sampling method by considering the trade-off between diversity and risk at each decoding step, based on our collected prefix tree which preserves the context of a full sentence. Our work offers a comprehensive comparison of existing truncation sampling methods and serves as a practical user guideline for their parameter selection. Our code is available at anonymized repository.

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Jiang et al., 2023; Team et al., 2023) have demonstrated exceptional performance across a variety of applications, and the reliability of decoding strategies has become a critical concern. Previous works have revealed that likelihood-maximization such as beam search (Fan et al., 2018; Holtzman et al., 2020; Welleck et al., 2020; Meister et al., 2022) produces degenerate text which contains repetitive loops and incoherent context, particularly in open-ended tasks. Therefore, sampling-based decoding strategies, e.g., Top-p (Holtzman et al., 2020) and Top-k sampling (Radford et al., 2018; Fan et al., 2018), have been widely adopted. The balance between diversity and quality of the generated text could be adjusted by tuning the temperature and truncation position to some extend, but requires non-trivial trial and error.

Recent studies (Basu et al., 2021; Zhu et al., 2024; Hewitt et al., 2022; Meister et al., 2023) proposed adaptive tail truncation mechanisms based on different criteria or assumptions, which maintain an allowed set of tokens with a flexible size according to the given prefix. To validate the effectiveness of a sampling method, they are often compared through extrinsic evaluation based on open-ended text generation applications. For example, story generation (Fan et al., 2018) and document continuation (Merity et al., 2017). Various metrics (Welleck et al., 2020; Meister et al., 2023; Pillutla et al., 2021; Gao et al., 2021) have been adopted to consider different aspects of the generated text.

We reveal two underlying issues in the current evaluation, which hinder the assessment of a method's significance in real-world applications:

- **The improvement of one method over another might be simply due to a better tuned parameter for the targeted task**: the performance of sampling methods is sensitive to their parameters, and parameter sweep is often operated on a extremely sparse grid due to the high computation cost. This is especially problematic considering the non-linear dependency between performance and parameters.

- **Users are agnostic to the optimal parameters in real-world applications**: Practically speaking, users often pick parameters based on their own need for the compromise between diversity and quality, after a few tryouts. There exists no universal optimal paramters in different scenarios and users are agnostic to the optimal parameters for their own tasks.

The above issues exactly indicate the need for an evaluation that allows for estimating the theoretical capacity of a truncation sampling method (how

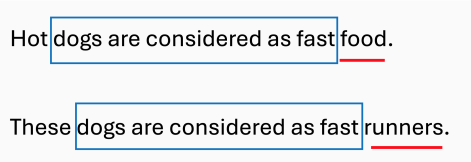


Figure 1: N-gram models tend to overestimate the data support size given a prefix (marked by a red line) due to limited window size (marked with a blue window).

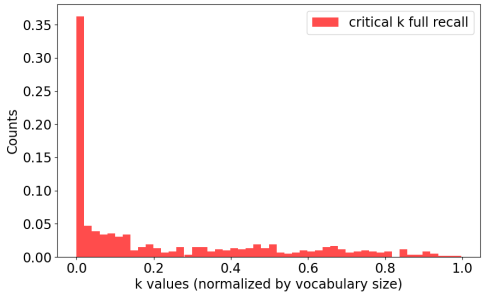


Figure 2: Histogram of the estimated optimal truncation values for gpt2-xl, which achieve exactly full recall of data support given different prefixes.

well it adapts to the variation in data supports given different prefixes), independent of hyperparamter tuning. Moreover, the second issue additionally highlights the need to identify the sweet spots of existing sampling methods, which could serve as a user guideline for practitioners.

In light of the above analysis, we propose a systematic way to assess the inherent adaptability of a sampling method. First, we rearrange Wikipedia-English [1] data into a word-level prefix tree, known as a Trie (Fredkin, 1960; Ghasemi et al., 2019). It is noteworthy that a n-gram Trie (Jurafsky, 2000) tends to overestimate the data support size given a prefix (Bengio et al., 2000), as shown in Figure 1. In a similar spirit to (Ding et al., 2024), we construct the prefix tree with only sentence-starting n-grams to preserve full sentence context, called Context-Preserving Trie (CP-Trie).

Given the CP-Trie, we are able to estimate the theoretical capacity of a sampling method, by examining the amount of tokens within and out of the data support with varying truncation parameter values. As shown in Figure 2, the truncation positions, which exactly cover the full data supports, vary drastically given different prefixes and Top-k sampling could be regarded as a baseline method with zero adaptability. Therefore, an adaptive truncation method is supposed to better follow such a variation, so that improved diversity can be achieved without harming the quality.

In summary, our contributions are as follows:

- We establish an intrinsic evaluation benchmark based on the collected CP-Trie, which allows for estimating the theoretical capacity of different sampling methods via thoroughly designed diversity and stability metrics.
- We conduct a comprehensive comparison of existing sampling approaches, which serves as a guideline for choosing a method and its parameter in real-world applications.
- We reveal that sampling-based decoding methods are underestimated in the existing study (Shi et al., 2024a) due to the difficulty in parameter selection, highlighting the merit of our evaluation protocol.

[1] https://dumps.wikimedia.org/

2 Related Work

In this section, we summarize recent sampling decoding strategies, along with common benchmarks and metrics for open-ended text generation.

2.1 Sampling-based Decoding Methods

Vanilla sampling suffers from the risk of obtaining incoherent tokens; thus, truncation of the tail distribution has been heavily discussed, e.g., Top-k (Radford et al., 2018; Fan et al., 2018) and Top-p sampling (Holtzman et al., 2020). However, a fixed k or p is problematic when considering the high dynamic range of next reasonable tokens, as pointed out in more recent studies on adaptive sampling methods: Mirostat (Basu et al., 2021) is proposed based on Zipf statistics and the assumption of a steady perplexity during generation. Hewitt et al. (2022) introduce η-sampling which dismisses the tokens with low probabilities in the tail of the predicted distribution based on absolute and relative thresholds. Locally Typical Sampling (Meister et al., 2023) assumes that the generated text should retain a similar entropy rate to that of human-generated text. Adaptive Decoding (Zhu et al., 2024) proposes to keep the entropy of the truncated distribution close to the original entropy. Although these approaches have been demonstrated to be effective, their performance is highly dependent on the curated truncation parameters and the limited exemplar text.

2.2 Evaluation of Sampling-based Decoding

Common benchmarks include story generation with WritingPrompts dataset (Fan et al., 2018), document continuation with WikiText-103 dataset (Merity et al., 2017) and abstractive summarization on the CNN/DAILYMAIL dataset (Nallapati et al., 2016). These benchmarks suffer from the problem of limited exemplar text, which fails to capture the diverse nature of human language.

Statistical metrics are mostly based on n-gram statistics and focus on a single aspect, such as Repetition (Welleck et al., 2020), Diversity (Meister et al., 2023), Semantic coherence (Gao et al., 2021), Zipf's coefficient (Holtzman et al., 2020) (Unigram rank-frequency) and Self-BLEU (Zhu et al., 2018).

Exemplar-based metrics dominate the evaluation of sampling-based decoding methods. As observed by Fan et al. (2018); Holtzman et al. (2020), lower perplexity of the generated text does not necessarily indicate better quality. And Holtzman et al. (2020) suggested that the perplexity of the generated text should be close to that of the human text. MAUVE (Pillutla et al., 2021) takes the trade-off between precision and recall into account, by comparing the learnt distribution from a text generation model to the distribution of human-written text using divergence frontiers. Shi et al. (2024a) provides a comprehensive evaluation on a large collection of tasks, mostly relying on exemplar-based metrics. However, we reveal that such evaluation is affected by the biases in the curated parameters and limited exemplar text, and our evaluation method is shown to alleviate such an issue.

3 Revisiting Truncation Sampling

We begin by revisiting the formulation of truncation sampling, followed by identifying the unresolved challenges in evaluating truncation sampling methods.

3.1 Problem Formulation

Definition 3.1.

$$P_{trunc}(x_t|\boldsymbol{x}_{<t}) = \begin{cases} P_\theta(x_t|\boldsymbol{x}_{<t})/Z_{\boldsymbol{x}_{<t}} & x \in \mathcal{A}_{\boldsymbol{x}_{<t}} \\ 0 & \text{o.w.,} \end{cases} \tag{1}$$

where $\mathcal{A}_{\boldsymbol{x}_{<t}} \in \mathcal{V}$ denotes the allowed set of candidate next tokens at the t^{th} position, given a sequence of tokens $\boldsymbol{x}_{<t} = \{x_0, ..., x_{t-1}\}$ as prefix. $Z_{\boldsymbol{x}_{<t}} = \sum_{x \in \mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta(x_t|\boldsymbol{x}_{<t})$ is the renormalization term.

Given the Context-Preserving Trie of a reference dataset, we can compute the estimate of the optimal allowed set as follows :

Definition 3.2. Let $\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}$ be the allowed set after truncation given the prefix $\boldsymbol{x}_{<t}$. The **approximated optimal allowed set** $\mathcal{A}^*_{\boldsymbol{x}_{<t}}$ corresponds to the allowed set with the minimum size, while covering the full data support for the t^{th} token $\mathcal{D}_{\boldsymbol{x}_{<t}}$ based on the Trie. It is the solution to the following objective function:

$$\begin{aligned} \mathcal{A}^*_{\boldsymbol{x}_{<t}} &= \min_\theta |\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}| \\ \text{s.t.} \quad & \mathcal{D}_{\boldsymbol{x}_{<t}} \subseteq \mathcal{A}_{\boldsymbol{x}_{<t,\theta}}. \end{aligned} \tag{2}$$

Note that the above definition is designed to exclude the risk of obtaining OOD tokens before the cutoff (Finlayson et al., 2024), because such type of risk is unsolvable by truncation and is rather determined by the capacity of the trained LLMs. However, such risk is less severe compared to that introduced by inappropriate truncation, since LLMs exhibit a significant capability in predicting the next token (Touvron et al., 2023; Achiam et al., 2023; Jiang et al., 2023; Team et al., 2023) and most OOD samples reside in the tail distribution.

3.2 Remaining Issues

We reveal three major issues in the evaluation of truncation sampling. We first summarize the problem of directly using probability as quality metric, then show that the choice of truncation parameter has a significant impact on the evaluation.

Unreliable Probability The probabilities of both the predicted and empirical distribution are not reliable for reflecting the quality of a text.

- Higher likelihood does not necessarily imply higher quality of the generated text (Fan et al., 2018; Holtzman et al., 2020; Nandwani et al., 2023; Wang and Zhou, 2024).
- Word frequencies are average statistics across various topics, and the optimal probabilities or ranking of each next token is ill-posed.
- Empirical distribution suffers from the sparsity issue (Shareghi et al., 2019; Li et al., 2016; Jurafsky, 2000) of the N-gram models.

Parameter Sensitivity We highlight the complexity and biases in parameter selection: Top-k and Top-p have constant upper bounds, i.e., the vocabulary size $|\mathcal{V}|$ and 1, respectively. In contrast, the

upper bounds of η-sampling and adaptive sampling are dependent on LLM's predicted distribution, because they truncate the tail distribution based on the likelihood of tokens and the slope of Min-Max scaled entropy, respectively. The importance of identifying the effective ranges of such parameters is also reflected in the authors' choice of numeral digit for their parameters. For example, ΔConf is set to 0.0005 in Zhu et al. (2024) and ϵ is chosen from 0.0001, 0.0009 and etc in Hewitt et al. (2022). In comparison, the adopted p values for Top-p sampling are merely two digits after zero, such as 0.95. This shows the significance of identifying the sweet spots of different sampling methods.

4 Method

In this section, we derive our metrics for evaluating different sampling-based decoding strategies. The metrics are carefully designed to address the issues discussed in Section 3.2.

4.1 Probability-Independent Metrics

To circumvent the **unreliable probability** issue, we merely check whether the predicted next token is in or out of the data support. Specifically, we define **Recall** and **Risk** to quantify diversity and quality of a sampling method on a single node of CP-Trie:

Definition 4.1.

$$\text{Recall}_{\theta,t} = \text{Minimum}\left(\frac{|\mathcal{A}_{\boldsymbol{x}_{<t},\theta}|}{|\mathcal{A}^*_{\boldsymbol{x}_{<t}}|}, 1\right) \tag{3}$$

$$\text{Risk}_{\theta,t} = \text{Maximum}\left(\frac{|\mathcal{A}_{\boldsymbol{x}_{<t},\theta}|}{|\mathcal{A}^*_{\boldsymbol{x}_{<t}}|} - 1, 0\right) \tag{4}$$

$\mathcal{A}_{\boldsymbol{x}_{<t},\theta}$ is dependent on the parameter selection for truncation, e.g., k value in Top-k sampling. When the allowed set is smaller than the approximated optimal allowed set after truncation, Recall is smaller than one and Risk is regarded as zero. With further increased size of the allowed set, Recall reaches one but Risk emerges. Since the sizes of reasonable sets vary drastically for different prefixes, it is not possible to always retain the approximated optimal allowed set with a predefined parameter. In this case, we reveal that the adaptability w.r.t. the varying size of data support of a sampling method indeed determines its effectiveness in real-world application.

More importantly, our evaluation does not rely on the empirical probability, which is biased and inaccurate due to limited dataset size or context window size. However, the tokens which appear in the dataset could be confidently regarded as reasonable, regardless of their actual probabilities. In addition, considering that temperature could change the flatness of distribution arbitrarily, we adopt ratio of token counts instead of probability mass to make the evaluation independent of temperature tuning and exemplar text. For a detailed discussion with supporting examples, please refer to Appendix A.2.

4.2 Tuning-Independent Evaluation

To eliminate the huge impact of **Parameter Sensitivity** issue on fair evaluation, we adopt **Average Recall (AR)** at an average Risk and **Risk Standard Error (RSE)** at an average Risk to quantify **diversity** and **stability** of a sampling method across N nodes of CP-Trie, respectively:

Definition 4.2.

$$\begin{aligned}
&\text{AR}_{\text{Risk}-0.1} = \frac{1}{N}\sum_{i=1}^{N}\text{Recall}^{(i)}_{\theta,t} \\
&\text{RSE}_{\text{Risk}-0.1} = \tfrac{1}{N}\sqrt{\textstyle\sum_{i=1}^{N}(\text{Risk}^{(i)}_{\theta,t} - \tfrac{1}{N}\sum_{i=1}^{N}\text{Risk}^{(i)}_{\theta,t})^2} \\
&\text{s.t.} \quad \frac{1}{N}\sum_{i=1}^{N}\text{Risk}^{(i)}_{\theta,t} = 0.1,
\end{aligned} \tag{5}$$

where the superscript (i) denotes the i^{th} node in the evaluation set of nodes on the prefix tree. Analogously, a family of critical values such as $\text{AR}_{\text{Risk}-0.5}$ can be easily defined.

Since θ is now determined by the given average Risk, the diversity metric reflects the genuine capacity of a sampling method regardless of parameter tuning. This allows for a fair comparison of different sampling methods, especially considering their drastically different effective ranges, as mentioned in Section 1 and Section 3.2.

5 Experiment

In this section, we conduct evaluation of existing sampling-based decoding approaches on our collected EnWiki CP-Trie dataset. We aim to estimate the inherent adaptability of sampling-based methods and the results could be used as references for the application of LLMs in open-ended tasks.

5.1 Data Collection

We construct our Trie data based on the English subset of Wikipedia dataset, named EnWiki CP-Trie. As shown in Figure 3, all possible words

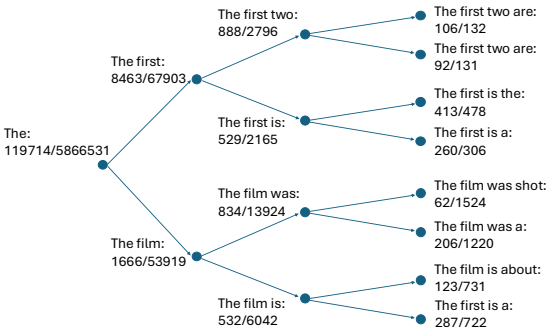


Figure 3: Illustration of the EnWiki CP-Trie. For brevity, only two child nodes are shown at each depth. The number at the left side of the slash symbol refers to the branching factor at the current node, and the number at the right side refers to the total number of leaves of the sub-tree with the current node as the root node.

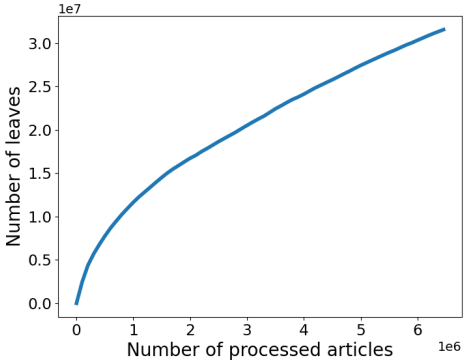


Figure 4: The total number of leaves on the CP-Trie against the total number of processed articles.

that appear after a given prefix in the dataset are treated as child nodes, with their preceding word regarded as the parent node. Starting from "Begin of Sequence" and collecting the child nodes recursively, we are able to transform the full dataset into a single prefix tree. We elaborate the main design choices in the following:

Basic Unit. It is possible to split the datasets into articles, paragraphs, sentences or n-grams. Constructing a tree based on articles or paragraphs may require more data than the training data of LLMs to guarantee an adequate number of branches (because LLMs lean to interpolate), whereas the construction based on n-grams suffers from poor contextual information and is heavily biased towards common tuplets of n tokens regardless of the context. Therefore, we adopt sentence as the basic unit, which guarantees a coherent context at sentence-level and requires much fewer data than training. It is noteworthy that a n-gram Trie (Jurafsky, 2000) tends to overestimate the data support size given a prefix (Bengio et al., 2000), due to the loss of information outside the contextual window, as shown in Figure 1.

Filtering. To avoid invalid words or rare proper names which are unreasonable for the model to predict, we exclude the sentences containing such words by checking their presence in the WORD LIST dataset, which is available on the website [2]. It contains 354986 words in total and explicitly excludes proper names and compound words. Section titles are also excluded, which are often incomplete sentences with poor contextual information.

Statistics. Wikipedia-English dataset contains $6,458,670$ articles, which result in EnWiki CP-Trie with $31,557,359$ leaves, see Figure 4.

Storage. The prefix tree is implemented as a nested dictionary and saved in JSON format. Since each lookup at any depth has constant complexity, the retrieval is highly efficient. Moreover, the dictionary is easily extendable if extra data are needed for a more accurate estimation of the full data support.

5.2 Evaluation Setup

Baselines. Our evaluation includes Top-k sampling (Radford et al., 2018; Fan et al., 2018), Top-p sampling (Holtzman et al., 2020), η-sampling (Hewitt et al., 2022), Adaptive sampling (Zhu et al., 2024) and Mirostat (Basu et al., 2021) into comparison.

Evaluation Data. To guarantee a tight lower bound of the ideal data support given different prefixes, we first sort the sub-nodes according to their total number of leaves at each depth, then we select the top 10 sub-trees with different sentence starting tokens for evaluation. Moreover, we keep the top 2 child nodes at each depth till depth 6, since the empirical data support becomes less adequate at large depth. This results in an evaluation set of 593 prefixes with varying lengths in total.

Evaluation Metrics. We measure the improvement in **diversity** via the increase of **Average Recall(AR)** at an average Risk, and the improvement of the **stability** at each decoding step in the auto-regressive process via the decrease of **Risk Standard Error (RSE)** at an average Risk. We adopt **AR** and **RSE** at average Risks of 1, 5 and 15 for comparison, representing low, medium, and high-risk regions, respectively.

LLMs. To ensure that the conclusion generalizes to different models, we adopt Llama (Touvron et al., 2023; Dubey et al., 2024) family, Mistral (Jiang

[2] word-list dataset homepage

et al., 2023, 2024) family and GPT-2-XL (Radford et al., 2019) for comparison.

Tokenization. Since different LLMs are trained with different encoding methods, the evaluation has to be independent of the encoding methods. We solve this issue by constructing the CP-Trie with either a word or punctuation. For example, if the predicted next token corresponds to “sec", which is a part of the in-distribution word "section", then we regard this as a correct prediction. The second part “tion" is regarded as a hidden child node and is skipped in the evaluation.

Parameter Search. We apply grid search to determine the corresponding parameters of different sampling methods for each average Risk. To address the highly non-linear dependency between the sampling methods and their truncation parameters, we employ an efficient coarse-to-fine grid search strategy: the number of grids is initially set to 2000. If a parameter results in an average Risk within ±0.1 of the target value, it is considered a feasible solution. Otherwise, an additional grid search is performed within a smaller interval until a feasible solution is found, based on the initial search results. The grids are determined using Llama3-70B and are applied consistently across all models. As shown in Table 7, almost all the deviations in the average Risks are much smaller than 0.1, demonstrating the robustness of our strategy.

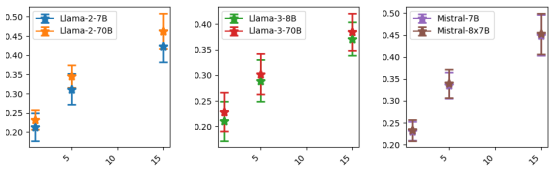
Implementation. Our implementation mainly relies on Pytorch (Paszke et al., 2017), HuggingFace (Wolf et al., 2020) and OpenAI API [3] library. We implement a truncation sampling method ourselves if the official implementation is unavailable. For all methods, the minimum size of the allowed set is set to 1 to prevent breaking the sampling process.

5.3 Comparison at Different Average Risks

In this section, we conduct a comprehensive study of different truncation sampling methods at different average Risks. As discussed in Section 4.2, this allows for a fair comparison which is independent of parameter tuning. Moreover, we provide the corresponding parameters for each truncation sampling method at different average Risks, which could serve as a user reference for these methods.

As can be seen in Table 1, different truncation sampling methods are compared at the average Risk of 1, 5, and 15 respectively. As discussed in Section 4.1, our defined risk and recall metrics explicitly exclude the source of risk induced by a LLM’s capacity by design, thus similar parameter values correspond to the same risk level for most sampling methods across various model types and sizes. This exactly showcases the advantage of our evaluation being tuning-independent and sustainable to the rapid update of LLMs. Among the evaluated methods, Eta-sampling (Hewitt et al., 2022) is sensitive to the changes of model type and size, which might hinder its practical significance especially at a low risk level.

Regarding diversity, i.e., the average recall at the same average Risk, Adaptive sampling (Zhu et al., 2024) and Mirostat (Basu et al., 2021) are the best and second performers, which consistently outperform the Top-k baseline by a considerable margin. Top-p mostly exhibits inferior recall comparing to the Top-k baseline, so does Eta-sampling at the average Risk of 1. As for the stability represented by standard error of Risks, Top-k sampling reaches the best scores in most cases. In comparison, Adaptive sampling and Mirostat deliver comparable standard error of risks to Top-k sampling, whereas Top-p sampling and Eta-sampling are again inferior. Considering both diversity and stability, Adaptive sampling and Mirostat are the top 2 adaptive methods to be recommended, whereas Top-p sampling shall be the last two methods to be considered.



(a) Llama-2 family. (b) Llama-3 family. (c) Mistral family

Figure 5: Comparing the average Recalls at given average Risks using different model sizes.

We also show in Figure 5 that larger models of the same family have higher average recall at the same risk level comparing to the smaller ones. This conforms to the fact that larger models better captures the human text distribution. Please note that our metrics does not allow a direct comparison between different model families, mainly due to their different vocabulary sizes and tokenizers, e.g., Llama-3 has a 128,256 vocabulary size, while Llama-2 has only 32, 000 vocabulary size. Moreover, our metrics also explicitly exclude the source of risk within the optimal allowed set, which is heavily dependent on a LLM’s capacity.

[3] https://pypi.org/project/openai/

Model	Method	Avg. Risk 1			Avg. Risk 5			Avg. Risk 15		
		Parameter	RSE ↓	AR ↑	Parameter	RSE ↓	AR ↑	Parameter	RSE ↓	AR ↑
GPT2-XL	Adaptive	9.5e-4	0.006	**0.252**	1.1e-4	0.679	**0.339**	2.5e-05	2.241	**0.413**
	Mirostat	4.425	**0.005**	0.236	5.9475	0.717	0.326	6.76	2.501	0.401
	Top-k	15	0.006	0.220	64	**0.613**	0.290	184	**1.781**	0.340
	Eta	0.318	0.013	0.198	0.011	1.484	0.301	0.001	4.261	0.404
	Top-p	0.5705	0.015	0.170	0.746	2.129	0.240	0.8555	6.210	0.338
Llama-2-7b	Adaptive	1.1e-3	0.154	**0.257**	1.4e-4	0.856	**0.364**	3.1e-5	2.966	0.470
	Mirostat	4.253	0.133	0.236	5.82	0.650	0.349	6.628	2.286	**0.474**
	Top-k	14	**0.126**	0.226	61	**0.587**	0.296	177	**1.722**	0.369
	Eta	0.512	0.563	0.192	0.023	2.599	0.297	0.002	6.531	0.407
	Top-p	0.54	0.529	0.156	0.7665	2.331	0.254	0.9	6.208	0.400
Llama-2-70b	Adaptive	0.0011	0.142	**0.269**	1.2e-4	0.796	0.374	2.3e-5	2.697	0.485
	Mirostat	4.16	0.135	0.238	5.7875	0.684	0.353	6.67	2.125	0.478
	Top-k	14	**0.128**	0.232	60	**0.583**	0.307	174	**1.712**	0.375
	Eta	0.092	0.304	0.236	0.003	1.590	**0.378**	2.1e-4	4.243	**0.510**
	Top-p	0.6535	0.475	0.189	0.8465	2.136	0.316	0.9395	5.522	0.468
Llama-3-8B	Adaptive	1.1e-3	0.167	**0.260**	1.7e-4	0.787	**0.343**	3.7e-5	2.685	**0.418**
	Mirostat	4.24	0.139	0.230	5.8175	0.804	0.318	6.693	2.630	0.393
	Top-k	14	**0.128**	0.228	59	**0.576**	0.290	172	**1.701**	0.346
	Eta	0.673	0.445	0.181	0.029	2.112	0.271	0.002	6.009	0.373
	Top-p	0.5395	0.451	0.154	0.736	2.061	0.224	0.855	5.770	0.326
Llama-3-70B	Adaptive	1.1e-3	0.137	**0.263**	1.4e-4	0.787	**0.353**	3.16e-5	2.778	**0.424**
	Mirostat	4.21	0.138	0.230	5.91	0.708	0.332	6.84	2.193	0.417
	Top-k	14	**0.127**	0.230	60	**0.581**	0.295	173	**1.695**	0.352
	Eta	0.37	0.137	**0.263**	0.014	2.231	0.295	0.001	6.265	0.398
	Top-p	0.5695	0.502	0.158	0.758	2.386	0.237	0.8705	6.685	0.332
Mixtral-7B	Adaptive	0.00105	0.152	**0.260**	1.2e-4	0.809	0.364	2.2e-5	2.757	0.466
	Mirostat	4.1825	0.141	0.236	5.8125	0.721	0.345	6.71	2.213	0.468
	Top-k	14	**0.126**	0.224	62	**0.596**	0.297	181	**1.759**	0.364
	Eta	0.075	0.307	0.243	0.003	1.542	**0.368**	1.96e-4	4.712	**0.505**
	Top-p	0.6565	0.539	0.194	0.8375	2.476	0.303	0.9315	6.315	0.447
Mixtral-8x7B	Adaptive	0.00105	0.148	**0.265**	1.1e-4	0.798	0.372	2.1e-5	2.802	0.476
	Mirostat	4.2775	0.143	0.238	5.845	0.710	0.346	6.6875	2.213	0.461
	Top-k	15	**0.134**	0.229	63	**0.598**	0.301	183	**1.757**	0.366
	Eta	0.087	0.335	0.241	0.003	1.822	**0.375**	2.15e-4	4.922	**0.506**
	Top-p	0.6505	0.535	0.192	0.8375	2.423	0.303	0.9325	6.139	0.456

Table 1: Risk Standard Error (RSE, indicating stability) and Average Recall (AR, indicating diversity) of different truncation sampling methods at different average Risks using different models. The corresponding parameter of each method at an average risk level is also provided. The best and worst scores are marked in bold and underlined, respectively. For more detailed results, please refer to Appendix A.1.

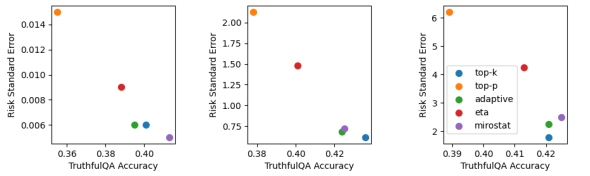
Methods	Mean(std) Accuracy ↑		
	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Greedy	0.338		
Naïve	0.421(0.004)		
Top-k	0.401(0.010)	**0.436**(0.008)	0.421(0.010)
Top-p	0.355(0.013)	0.378(0.011)	0.389(0.012)
Adaptive	0.395(0.012)	0.424(0.011)	0.421(0.009)
Eta	0.388(0.005)	0.401(0.013)	0.413(0.026)
Mirostat	0.413(0.010)	0.425(0.013)	**0.425**(0.009)

Table 2: Evaluation on the TruthfulQA benchmark under the open-ended generation setup. Naive sampling refers to sampling without truncation. The best and worst scores are marked in bold and underlined, respectively. For more details, please refer to Appendix A.1.

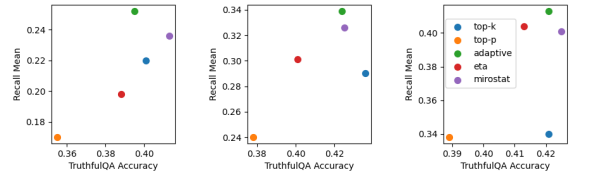
5.4 Validation on TruthfulQA Benchmark

Although our evaluation protocol is grounded by the thorough design process with reasonable simplifications, we would like to verify its effectiveness in the real-world scenario using the TruthfulQA Benchmark (Lin et al., 2021). The evaluation results using gpt2-xl are shown in Section 5.3. For all the methods other than greedy decoding, we run 3 times at each average risk level and report the mean and standard deviation (parenthetical value).

It can be observed that greedy decoding falls far behind sampling-based decoding strategies, which conforms to the issue of likelihood-oriented decoding discussed in Section 1, as well as the findings in recent studies (Cobbe et al., 2021; Wang et al., 2023; Wang and Zhou, 2024; Shi et al., 2024a). All the truncation sampling methods at the low risk level achieves lower accuracy comparing to Naive sampling, due to the over-truncation of the decoding paths. At the average risk level of 5, all the truncation sampling methods slightly improve their own accuracy. Top-k sampling, Adaptive sampling and Mirostat also reach comparable or slightly higher accuracy in comparison to Naive sampling. However, further increased average risk level (means improved average recall and



(a) Correlation at Avg. Risk 1: −0.87 (b) Correlation at Avg. Risk 5: −0.92 (c) Correlation at Avg. Risk 15: −0.94



(d) Correlation at Avg. Risk 1: 0.83 (e) Correlation at Avg. Risk 5: 0.83 (f) Correlation at Avg. Risk 15: 0.50

Figure 6: The scatter plots of TruthfulQA accuracy against risk standard error (first row) and recall mean (second row) at different average Risks.

thus diversity) does not benefit the performance on TruthfulQA, which is plausible. Moreover, there exists a even stronger correlation between Risk SE (Standard Error of Risks) and TruthfulQA accuracy, validating the importance of stability when evaluating an adaptive decoding method. The strong correlation between TruthfulQA accuracy and our proposed average recall as well as standard error of risks at different average Risks validate the soundness and effectiveness of our evaluation method.

6 Revisiting Existing Evaluation

In this section, we revisit the recent study (Shi et al., 2024a) by comparing sampling-based decoding methods at the same average Risks. We adopt the official implementation of Shi et al. (2024a). Following their setups, we adopt Llama-2-7B on MBPP (Austin et al., 2021), HumanEval (Austin et al., 2021) and GSM8K (Cobbe et al., 2021) to evaluate coding and math problem solving performance. Mean and standard deviation for three runs are reported in Table 3, Table 4 and Table 5, respectively.

For all the three tasks, Mirostat does not perform well in general, probably because it is based on the Zipf-law of natural language and thus not suitable for code and math tasks. Notably, our greedy decoding baseline achieves significantly lower result than reported by Shi et al. (2024a) on HumanEval. Our results should be plausible, because the instruction tuned Llama-2-7B only achieves 7.9 according to Meta-Llama Github[4] .

[4] https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

While their study concludes that deterministic methods outperform sampling methods across most tasks, our evaluation reveals that sampling methods are indeed underestimated. in contrast to the conclusion in Shi et al. (2024a), all the sampling-based decoding methods could achieve better performance than greedy decoding on HumanEval in Table 4. In addition, Top-p and eta sampling also beat greedy decoding at a low average Risk on GSM8K in Table 5. This observation underscores the challenges in parameter selection for sampling-based decoding, which is effectively addressed by our method.

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	19.70 (0.50)	21.00 (2.30)	20.50 (0.30)
Top-p	21.50 (1.30)	**21.10** (0.40)	**21.70** (0.70)
Mirostat	9.50 (0.30)	8.80 (2.00)	8.80 (0.40)
Eta	**22.10** (0.70)	19.10 (0.40)	19.70 (0.40)
Greedy		24.00	

Table 3: Pass@1 accuracy on MBPP. It is consistent to the observation by Shi et al. (2024a) that sampling methods are inferior to greedy decoding.

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	**5.68** (2.00)	5.08 (1.52)	6.50 (0.76)
Top-p	3.46 (1.05)	5.89 (0.76)	**6.52** (2.43)
Mirostat	3.25 (0.76)	4.27 (1.00)	4.68 (0.58)
Eta	2.64 (2.01)	**6.91** (1.04)	6.10 (1.32)
Greedy		2.44	

Table 4: Pass@1 accuracy on HumanEval. Sampling methods perform better with higher average Recalls and Risks.

Methods	Avg. Risk 1	Avg. Risk 5	Avg. Risk 15
Top-k	7.56 (5.39)	**11.90** (0.80)	**11.73** (0.57)
Top-p	**14.13** (0.47)	8.72 (6.18)	11.67 (0.11)
Mirostat	5.46 (0.47)	5.74 (0.64)	3.46 (2.10)
Eta	13.72 (0.46)	8.42 (5.54)	11.22 (0.75)
Greedy		13.19	

Table 5: Accuracy on GSM8K. Top-p and eta sampling outperforms greedy decoding at an average Risk of 1.

7 Conclusion

In this work, we propose an evaluation protocol to assess the trade-off between diversity and quality of truncation sampling methods for open-ended text generation. Our evaluation enjoys the merit of being independent of parameter tuning for the curated tasks. The evaluation results also serve as a user reference for different downstream tasks.

8 Limitations

In this work, we focus on the truncation sampling methods specially designed for the open-ended text generation scenario. There exist many related decoding strategies, which aim at improving different aspects of LLMs. For example, a line of decoding strategies is proposed to alleviate hallucination or improve the reasoning ability, e.g., Dola (Chuang et al., 2023), Context-aware decoding (Shi et al., 2024b), Contrastive decoding (O'Brien and Lewis, 2023) and etc. However, these methods are beyond the scope of sampling-based decoding in this study and thus not included in the discussion. Although our study is only based on text data in English for clarity, the dataset can be extended to include other languages in the future. Due to time and resource constraints, we did not include all existing sampling-based decoding methods, such as Locally Typical Sampling (Meister et al., 2023) and Min-P Sampling (Nguyen et al., 2024), in our comparison. However, our benchmark is publicly available, and we plan to continuously update it with evaluations of additional methods in the future.

9 Broader Impact

Our study on the capacity of sampling methods and their appropriate parameters for open-ended text generation may further promote the application of LLMs in creative industries. There exists a potential risk that our provided findings might be abused for generating harmful or fake information. However, our study itself is neutral and the mentioned risk is a general issue that LLMs face. We call for the attention on AI-Safety in the community.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

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

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in Neural Information Processing Systems (NeurIPS)*, 13.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *The Twelfth International Conference on Learning Representations (ICLR)*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *URL https://arxiv.org/abs/2110.14168*.

Hantian Ding, Zijian Wang, Giovanni Paolini, Varun Kumar, Anoop Deoras, Dan Roth, and Stefano Soatto. 2024. Fewer truncations improve language modeling. *International Conference on Machine Learning (ICML)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. 2024. Closing the curious case of neural text degeneration. *International Conference on Learning Representations (ICLR)*.

Edward Fredkin. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Chavoosh Ghasemi, Hamed Yousefi, Kang G Shin, and Beichuan Zhang. 2019. On the granularity of trie-based data structures for name lookups and updates. *IEEE/ACM Transactions on Networking*, 27(2):777–789.

John Hewitt, Christopher D Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. *Findings of the Association for Computational Linguistics: EMNLP*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *The curious case of neural text degeneration*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Bofang Li, Zhe Zhao, Tao Liu, Puwei Wang, and Xiaoyong Du. 2016. Weighted neural bag-of-n-grams model: New baselines for text classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 2022. On the probability-quality paradox in language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. *International Conference on Learning Representations (ICLR)*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*.

Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi, and Luis A Lastras. 2023. Pointwise mutual information based metric and decoding strategy for faithful generation in document grounded dialogs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:4816–4828.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ehsan Shareghi, Daniela Gerz, Ivan Vulic, et al. 2019. Show some love to your n-grams: A bit of progress and stronger n-gram language modeling baselines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024a. A thorough examination of decoding methods in the era of llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2024b. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

X Wang, J Wei, D Schuurmans, Q Le, E Chi, S Narang, A Chowdhery, and D Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations (ICLR)*.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems (NeurIPS)*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. *International Conference on Learning Representations (ICLR)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Huggingface's transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yiming Ai, and Rui Wang. 2024. Improving open-ended text generation via adaptive decoding. *International Conference on Machine Learning (ICML)*.

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 *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

A Appendix

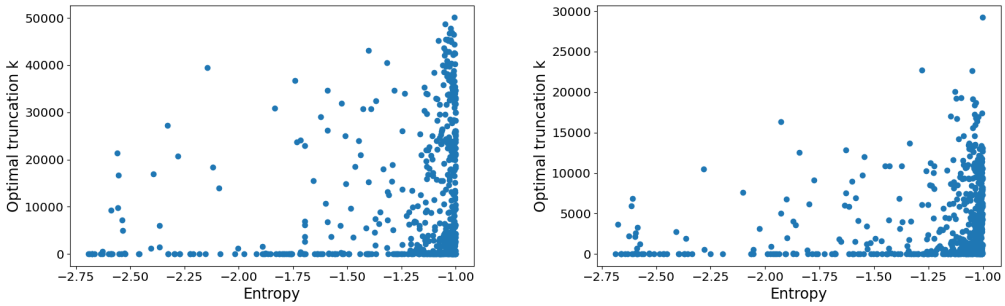
A.1 Complete Record of the Experiment Runs

Methods	Evaluation Runs									Mean/Std		
	Run 1 at average Risks			Run 2 at average Risks			Run 3 at average Risks			average Risks		
	1	5	15	1	5	15	1	5	15	1	5	15
Greedy Decoding	0.338											
Naive Sampling	0.420			0.426			0.416			0.421(0.004)		
Top-k Sampling	0.412	0.447	0.410	0.389	0.432	0.435	0.402	0.428	0.419	0.401(0.010)	0.436(0.008)	0.421(0.010)
Top-p Sampling	0.337	0.370	0.382	0.367	0.393	0.379	0.362	0.370	0.405	0.355(0.013)	0.378(0.011)	0.389(0.012)
Adaptive Sampling	0.403	0.416	0.433	0.403	0.416	0.419	0.378	0.440	0.411	0.395(0.012)	0.424(0.011)	0.421(0.009)
Eta Sampling	0.395	0.419	0.442	0.387	0.394	0.419	0.382	0.389	0.379	0.388(0.005)	0.401(0.013)	0.413(0.026)
Mirostat	0.424	0.417	0.430	0.399	0.443	0.433	0.415	0.414	0.412	0.413(0.010)	0.425(0.013)	0.425(0.009)

Table 6: Evaluation on the TruthfulQA benchmark. Since the GPT-3 API is no longer available, we use the by the authors recommended BLEURT accuracy for comparison under the open-ended generation setup.

The scores of the individual runs on TruthfulQA benchmark are recorded in Table 6, and the means and standard errors of recalls and risks at all average Risks are listed in Table 7. Note that due to a fixed amount of computation budget, we search the corresponding parameter value for each truncation sampling method till the average risk is close enough to the predefined value, thus resulting in the variations of the average risks. However, such variations are negligible given the minor differences.

Although Top-p sampling is indeed also adaptive regarding the truncation position, we show that Top-p sampling have a inherent limitation. When a larger portion of the probability mass is concentrated in the first few tokens (this often indicates smaller entropy), a fixed cumulative probability threshold will cut a longer tail off, and vice versa. However, there's merely a weak correlation between the entropy of the LLM's prediction and optimal truncation values, see Figure 7.



(a) The Pearson's correlation is 0.24777 for GPT2-XL. (b) The Pearson's correlation is 0.24784 for Llama-2-7B.

Figure 7: Scatter plots between the entropy values and optimal truncation values.

A.2 The Advantage of Probability-Independent Metrics

In this section, we explain the practical advantages of our proposed probability-independent recall and risk metrics. As can be seen in Figure 8, the empirical distribution aligns with the by gpt2-xl predicted distribution given the same prefix in general: most of the tokens which posses high likelihood in the prediction also has a high probability based on the word frequencies of our collected CP-Trie data. However, there exists two differences:

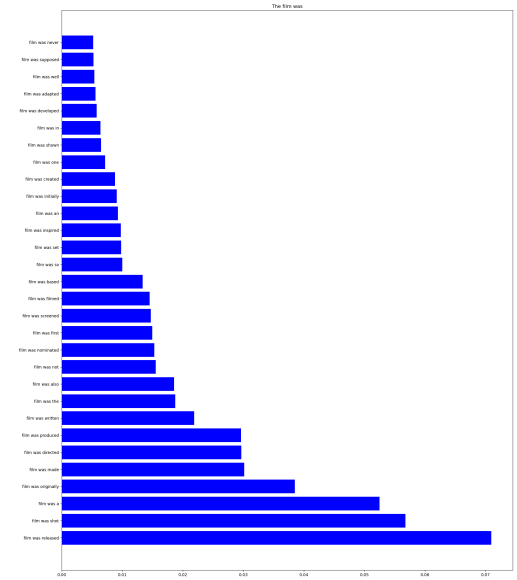
- Some tokens with high likelihood according to gpt2-xl have much lower probability according to the empirical distribution. The ranking of each tokens w.r.t. probability also differ in the two distributions.
- A few tokens which should be reasonable candidates (by manual check) have 0 probability according to the empirical distribution.

For the first issue, as discussed in Section 3.2, there exists no ideal probabilities for each token, and the discrepancy is not solvable by simply increasing the size of the data. For example, the "perfect" probabilities of the candidate tokens "with" and "at" are undefined and could even be regarded as equivalently important for open-ended text generation.

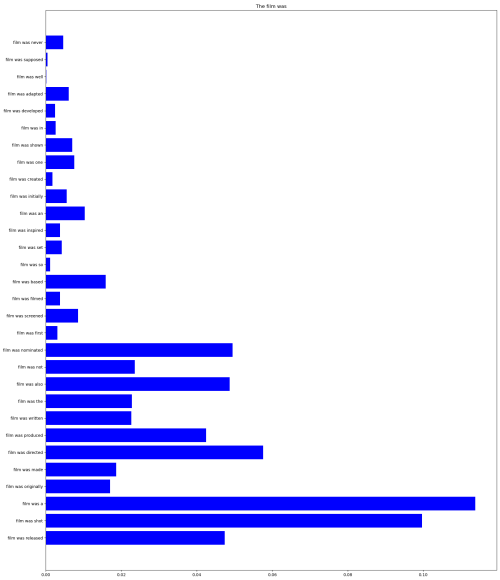
The second difference highlights the reliability of LLMs, i.e., the tokens which are assigned high likelihoods are in most cases reasonable. Note that we ignore the risk within the estimated optimal allowed set by design: All the tokens are counted as reasonable till the last token which has non-zero empirical probability, when they are arranged in a descending order according to the predicted probabilities. Thus these tokens with zero probabilities in the empirical distribution will not affect our evaluation of risk, making our method robust to noises and insufficient data support.

Method					GPT2-XL				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	15	1.029 (0.006)	0.220(0.0006)	64	5.040 (0.613)	0.290 (0.017)	184	14.983(1.781)	0.340 (0.018)
Top-p	0.5705	0.999 (0.015)	0.170 (0.0005)	0.746	5.011(2.129)	0.240 (0.015)	0.8555	15.022 (6.210)	0.338 (0.016)
Adaptive	9.5e-4	1.000 (0.006)	0.252 (0.0007)	0.00011	4.997 (0.679)	0.339(0.018)	2.5e-05	14.995 (2.241)	0.413 (0.018)
Eta	0.318	1.000 (0.013)	0.198 (0.0005)	0.011	4.945 (1.484)	0.301 (0.016)	0.001	14.998 (4.261)	0.404 (0.017)
Mirostat	4.425	0.999 (0.005)	0.236 (0.0007)	5.9475	5.001 (0.717)	0.326 (0.018)	6.76	14.982 (2.501)	0.401 (0.018)
Method					Llama-2-7b				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	0.986 (0.126)	0.226 (0.016)	61	4.987 (0.587)	0.296 (0.017)	177	14.961 (1.722)	0.369 (0.018)
Top-p	0.54	0.999 (0.529)	0.156 (0.012)	0.7665	4.990 (2.331)	0.254 (0.015)	0.9	14.989 (6.208)	0.400 (0.016)
Adaptive	0.0011	1.051 (0.154)	0.257 (0.016)	0.00014	4.991 (0.856)	0.364 (0.017)	3.1e-5	14.995 (2.966)	0.470 (0.017)
Eta	0.512	1.000 (0.563)	0.192 (0.014)	0.023	5.007 (2.599)	0.297 (0.016)	0.002	13.487 (6.531)	0.407 (0.017)
Mirostat	4.253	1.000 (0.133)	0.236 (0.016)	5.82	4.993 (0.650)	0.349 (0.018)	6.628	15.022 (2.286)	0.474 (0.017)
Method					Llama-3-8B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.023 (0.128)	0.228 (0.016)	59	4.982 (0.576)	0.290 (0.017)	172	15.025 (1.701)	0.346 (0.018)
Top-p	0.5395	1.000 (0.451)	0.154 (0.013)	0.736	4.998 (2.061)	0.224 (0.014)	0.855	14.993 (5.770)	0.326 (0.016)
Adaptive	0.0011	1.133 (0.167)	0.260 (0.017)	0.00017	5.006 (0.787)	0.343 (0.018)	3.7e-5	15.007 (2.685)	0.418 (0.018)
Eta	0.673	1.000 (0.445)	0.181 (0.014)	0.029	5.009 (2.112)	0.271 (0.016)	0.002	15.012 (6.009)	0.373 (0.017)
Mirostat	4.24	1.001 (0.139)	0.230 (0.016)	5.8175	5.001 (0.804)	0.318 (0.018)	6.6925	14.996 (2.630)	0.393 (0.018)
Method					Llama-3-70B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.014 (0.127)	0.230 (0.016)	60	5.038 (0.581)	0.295 (0.017)	173	15.024 (1.695)	0.352 (0.018)
Top-p	0.5695	1.001 (0.502)	0.158 (0.013)	0.758	4.999 (2.386)	0.237 (0.015)	0.8705	14.960 (6.685)	0.332 (0.016)
Adaptive	0.0011	1.004 (0.137)	0.263 (0.017)	0.00014	5.013 (0.787)	0.353 (0.018)	3.16e-5	14.986 (2.778)	0.424 (0.018)
Eta	0.37	1.004 (0.137)	0.263 (0.017)	0.014	5.032 (2.231)	0.295 (0.016)	0.001	15.076 (6.265)	0.398 (0.018)
Mirostat	4.21	1.001 (0.138)	0.230 (0.016)	5.91	5.001 (0.708)	0.332 (0.018	6.84	15.021 (2.193)	0.417 (0.018)
Method					Llama-2-70b				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	1.002 (0.128)	0.232 (0.016)	60	4.982 (0.583)	0.307 (0.017)	174	14.964 (1.712)	0.375 (0.018)
Top-p	0.6535	0.999 (0.475)	0.189 (0.013)	0.8465	4.988 (2.136)	0.316 (0.016)	0.9395	15.019 (5.522)	0.468 (0.016)
Adaptive	0.0011	1.000 (0.142)	0.269 (0.017)	1.2e-4	4.995 (0.796)	0.374 (0.017)	2.3e-5	15.007 (2.697)	0.485 (0.017)
Eta	0.092	1.002 (0.304)	0.236 (0.015)	0.003	5.057 (1.590)	0.378 (0.017)	0.00021	15.001 (4.243)	0.510 (0.017)
Mirostat	4.16	1.001 (0.135)	0.238 (0.016	5.7875	5.004 (0.684)	0.353 (0.018)	6.67	14.991 (2.125)	0.478 (0.017)
Method					Mixtral-8x7B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	15	1.028 (0.134)	0.229 (0.016)	63	4.978 (0.598)	0.301 (0.017)	183	14.967 (1.757)	0.366 (0.018)
Top-p	0.6505	1.000 (0.535)	0.192 (0.014)	0.8375	5.007 (2.423)	0.303 (0.015)	0.9325	14.966 (6.139)	0.456 (0.016)
Adaptive	0.00105	1.000 (0.148)	0.265 (0.017)	0.00011	4.994 (0.798)	0.372 (0.018)	2.1e-5	15.014 (2.802)	0.476 (0.017)
Eta	0.087	1.001 (0.335)	0.241 (0.015)	0.003	5.061 (1.822)	0.375 (0.017)	0.000215	14.991 (4.922)	0.506 (0.017)
Mirostat	4.2775	1.000 (0.143)	0.238 (0.016)	5.845	4.995 (0.710)	0.346 (0.018)	6.6875	14.998 (2.213)	0.461 (0.018)
Method					Mistral-7B				
	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
Top-k	14	0.965 (0.126)	0.224 (0.016)	62	4.968 (0.596)	0.297 (0.017)	181	15.006 (1.759)	0.364 (0.018)
Top-p	0.6565	1.001 (0.539)	0.194 (0.014)	0.8375	4.996 (2.476)	0.303 (0.016)	0.9315	15.038 (6.315)	0.447 (0.016)
Adaptive	0.00105	1.001 (0.152)	0.260 (0.016)	0.000115	4.993 (0.809)	0.364 (0.018)	2.2e-5	14.999 (2.757)	0.466 (0.017)
Eta	0.075	0.997 (0.307)	0.243 (0.015)	0.003	4.640 (1.542)	0.368 (0.017)	0.000196	15.009 (4.712)	0.505 (0.017)
Mirostat	4.1825	1.000 (0.141)	0.236 (0.016)	5.8125	4.999 (0.721)	0.345 (0.018)	6.71	14.978 (2.213)	0.468 (0.018)

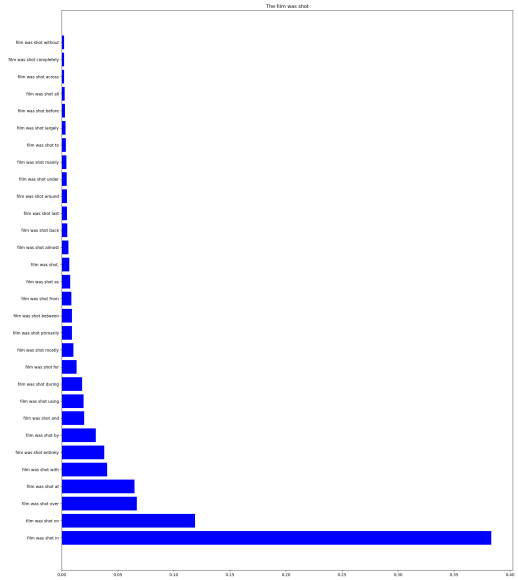
Table 7: Critical Parameters of different truncation sampling methods at different average Risks using different models.



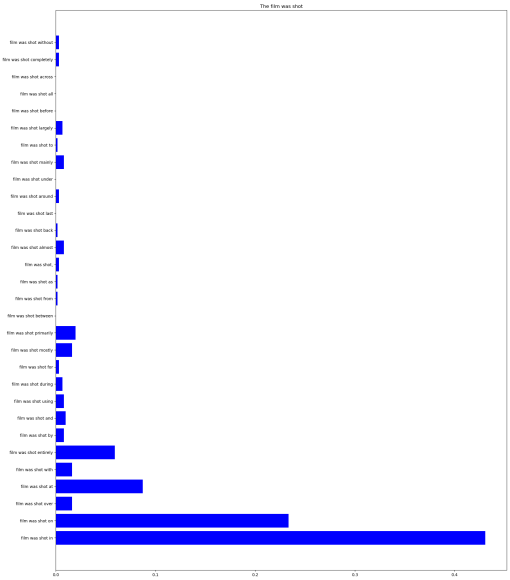
(a) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding likelihood given the prefix "The film was"



(b) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding empirical probability given the prefix "The film was".



(c) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding likelihood given the prefix "The film was shot".



(d) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding empirical probability given the prefix "The film was shot".

Figure 8: Comparing the probabilities predicted by gpt2-xl and calculated using the word frequencies based on our collected CP-Trie data.