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, which target a balance between diversity and quality via temperature tuning and tail truncation (e.g., top-k and top-p sampling). Considering the high dynamic range of the candidate next-tokens given different prefixes, recent studies propose to adaptively truncate the tail of LLM's predicted distribution. Although improved results haven been reported with these methods on open-ended text generation tasks, the results are highly dependent on the curated truncation parameters and exemplar text. In this paper, we propose a systematic way to estimate the intrinsic 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 provides a comprehensive comparison between existing truncation sampling methods, as well as their recommended parameters as a guideline for users. 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 incredible performance in a variety of applications, and the reliability of decoding strategies has become a critical concern, especially where diverse and coherent samples are desired. 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 degenerated text which contains repetitive loops and incoherent context, particularly in the 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 that there exist two underlying issues in the current evaluation setup, which might hinder the assessment of a sampling method's practical significance in real-world applications:

- **The improvement of one method against another may be simply due to a better tuned parameter for the targeted exemplar text**: 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 few number of tryouts. There exits no universal optimal hyperparamters in different scenarios and users

are agnostic to the optimal hyerparameters for their own tasks.

The above issues exactly indicate the need for an evaluation that allows for estimating the theoretical capacity of a sampling method, independent of hyperparamter tuning. Moreover, the second issue additionally highlights the need for identifying the sweet spots of existing sampling methods, which could serve as a general guideline for parameter selection when applying a method.

Based on the above analysis, we propose a systematic way to assess the inherent adaptability of a sampling method in different contexts. First, we rearrange Wikipedia-English [1] data in the form of a word-level prefix tree structure, or the so-called Trie (Fredkin, 1960; Ghasemi et al., 2019). As shown in fig. 3, all possible words that appear after a given prefix in the dataset are collected together as the child nodes, and their preceding word is 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. It is noteworthy that a n-gram Trie (Jurafsky, 2000) tends to produce overestimated data support size given a prefix (Bengio et al., 2000), due to the lost of contextual information outside the truncation window, as shown in fig. 1. In a similar spirit to (Ding et al., 2024), we intentionally construct the prefix tree with only sentence-starting n-grams to preserve the context of a full sentence. thus we refer to the collected data as Context-Preserving Trie (CP-Trie).

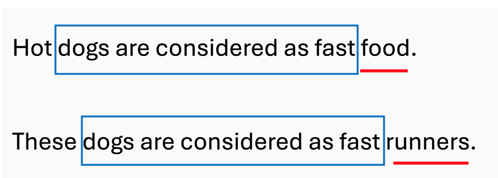


Figure 1: N-gram models tend to provide an overestimate of the data support size given a prefix (marked by a red line) due to the truncation of a full sentence (marked with a blue window).

Since the child nodes of the ancestral prefix define the data support, the recall of a sampling method w.r.t. data support is able to be computed. Though the training datasets of modern LLMs are significantly larger than Wikipedia dataset, such a data support serve as a reasonable lower-bound, especially when only sentence-level context are considered for evaluation, see section 4.1 for more details.

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

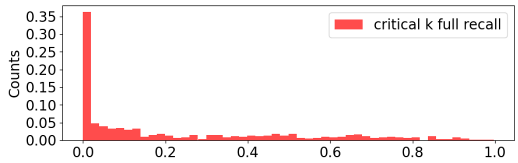


Figure 2: Histogram of the estimated optimal truncation values for gpt2-xl, which achieve exactly full recall of data support given different prefixes.

Given the context-preserving prefix tree, we are able to evaluate 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 can be seen in fig. 2, the optimal truncation position vary drastically given different prefixes and Top-k sampling could be regarded as a baseline method with zero adaptability. Therefore, an effective adaptive truncation mechanism is supposed to better follow such a variation, so that improved diversity can be achieved without harming the quality.

In summary, the contributions of our paper are as follows:

- We establish an intrinsic evaluation benchmark based on the Context-Preserving Prefix tree data, which allows for estimating the theoretical capacity of different sampling decoding methods via our proposed diversity and stability metrics.

- We conduct a comprehensive comparison of existing sampling approaches on our proposed benchmark, which serve as a guideline for choosing the truncation sampling methods and their corresponding parameter selection in real-world applications.

2 Related Work

In this section, we summarize the recently proposed sampling-based decoding strategies and the widely adopted benchmarks as well as 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 to alleviate such an issue, 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 the generated text to 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 empirically evaluated on open-ended text generation tasks with curated truncation parameters and exemplar text, there still lacks a comprehensive comparison of their adaptability in more general cases.

2.2 Evaluation of Sampling-based Decoding

Benchmarks The commonly adopted 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) (simple Unigram rank-frequency statistics) 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 doesn't necessarily indicate better quality. And Holtzman et al. (2020) suggested that the perplexity of the generated text is supposed to 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. A recent study (Shi et al., 2024a) provides a comprehensive evaluation on a large collection of tasks, which are mostly based on exemplar-based metrics.

3 Revisiting Truncation Sampling

3.1 Problem Formulation

Definition 3.1.

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

where $\mathcal{A}_{\boldsymbol{x}_{<t}} \in \mathcal{V}$ denotes the allowed set comprising candidate next-tokens for a given prefix, and $Z_{\boldsymbol{x}_{<t}} = \sum_{x \in \mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta(x_t|\boldsymbol{x}_{<t})$ is the renormalization term.

Given the Context-Preserving Trie of a reference dataset, we could 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 **optimal allowed set** $\mathcal{A}^*_{\boldsymbol{x}_{<t}}$ corresponds to the allowed set with the minimum size, while covering the full data support. It is the solution to the following objective function:

$$\begin{aligned} & \min_\theta |\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}| \\ s.t. \quad & \frac{|\mathcal{A}_{\boldsymbol{x}_{<t,\theta}} \bigcap \mathcal{D}_{\boldsymbol{x}_{<t}}|}{|\mathcal{D}_{\boldsymbol{x}_{<t}}|} = 100\%. \end{aligned} \tag{2}$$

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 the choice of truncation parameter has a significant impact on the evaluation. Finally, we reveal that a minor difference of Recall and Risk values may result in significant changes in diversity and quality of the generated text.

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

- Higher likelihood doesn't 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 assuming the optimal probabilities or the optimal ranking of each reasonable 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.

Impact of Parameter Selection 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 proposed 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. Our analysis also shows the significance of identifying the sweet spots of different sampling methods.

The Butterfly Effect Although the top few samples possess the most probability mass, we reveal that a minor change in the size of the allowed set at each decoding step could lead to major differences in the quality of the generated text, due to LLMs' auto-regressive nature. For example, the overall probability of at least one bad token to appear at the t^{th} position in a sequence of length T increases rapidly as T increases (due to the product operator):

$$P_\theta(\exists x_t \notin \mathcal{A}^*_{\boldsymbol{x}_{<t}}) = 1 - \prod_{t=1}^{T} P_\theta(x_t \in \mathcal{A}^*_{\boldsymbol{x}_{<t}} | \boldsymbol{x}_{<t}). \tag{3}$$

Similarly, if only 1% probability mass is assigned to additional tokens at each step, for a sequence of length T, there will be $1-0.99^T$ chances of obtaining extra diverse samples.

4 Method

In this section, we derive our method for evaluating different sampling-based decoding strategies. To circumvent the reliability issue of the probabilities we merely check whether the predicted next-token is in or out of the data support.

4.1 Probability-Independent Metrics

To quantify the diversity and quality of a sampling method based on CP-Trie, we define the **Recall** and **Risk** of a sampling method regarding a given prefix below:

Definition 4.1.

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

$$\text{Risk}_\theta = \text{Maximum}\left(\frac{|\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}|}{|\mathcal{A}^*_{\boldsymbol{x}_{<t}}|} - 1, 0\right) \tag{5}$$

$\mathcal{A}_{\boldsymbol{x}_{<t,\theta}}$ is dependent on the parameter selection for truncation, e.g., k value in top-k sampling. When the allowed set is smaller than the optimal one 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 optimal allowed set with a pre-defined 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.

Note that we ignore the risk of obtaining bad samples within the optimal allowed set, because such type of risk is unsolvable by truncation and is rather determined by the inherent capacity of the trained LLMs. However, such risk is less severe comparing 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 bad samples reside in the tail distribution.

More importantly, our evaluation doesn't 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 use ratio of token counts instead of probability mass to make the evaluation independent on temperature tuning and exemplar text. For a more detailed discussion with supporting examples, please refer to appendix A.2.

4.2 Parameter-Independent Evaluation

To eliminate the huge impact of parameter tuning on fair evaluation, we define the final diversity metric **Recall at a given Risk level** as follows:

Definition 4.2.

$$\text{Recall}_{\text{Risk}-0.1} = \text{Recall}_{\theta} \quad s.t. \quad \text{Risk}_{\theta} = 0.1 \tag{6}$$

Analogously, a family of critical values such as $\text{Recall}_{\text{Risk}-0.5}$ can be easily defined.

From the definition, it can be seen that such a metric is no longer dependent on the selection of θ parameter, thus it reflects the genuine capacity of a sampling method regardless of parameter tuning. This allows for a fair comparison between different sampling methods, especially considering their drastically different effective ranges, as mentioned in section 1 and section 3.2.

4.3 The Priority of Low Variance in Risk

Despite that more diverse text is desired for many generation tasks, the minimum requirement of the text to be coherent and reasonable is mostly prioritized in real-world applications. Besides the average recall at a given risk level, it is noteworthy that a stable adaptive truncation mechanism is also preferred , i.e., the variance of risks should be kept as low as possible at the given average risk level.

Conjecture 4.3. *At a given average risk level, the total amount of risk when generating a sequence of length T is reduced with decreased variance of the risks at each decoding step.*

The above conjecture can be mainly understood by the auto-regressive generation process of LLMs, as shown in eq. (3). The in-distribution probability dependent on the risk at each decoding step and is minimized when the product of the probabilities is maximized at a given average risk level. We could infer that the sum of the in-distribution probabilities is approximated determined at a given average risk level. For simplification, we assume their sum is unchanged. According to AM-GM inequality, the maximum of the product is achieved when each individual component is equal to each other. Roughly speaking, the less variance of the probability masses at each step, the larger their product and thus the smaller the total risk is.

5 Experiment

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

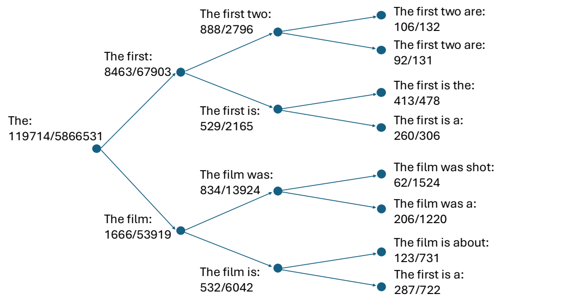


Figure 3: Illustration of the EnWiki CP-Trie. For brevity, only two child nodes are shown at each depth. The number at the left side of the slash symbol refers to the branching factor at the current node, and the number at the right side refers to the total number of leaves of the sub-tree with the current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

5.1 Data Collection

As mentioned above, the construction of the CP-Trie is a re-collection of an existing dataset. We apply the described procedure to the English subset of Wikipedia dataset and name the resulting dataset EnWiki CP-Trie. Although the core idea is straight-forward to understand, we elaborate the main design choices in the following:

Basic Unit There are many possible units for splitting the datasets into individual fragments, such as article, paragraph, sentence and n-grams. Constructing a tree based on articles or paragraphs may require a larger amount of 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 are heavily biased towards common tuplets of n tokens regardless of the context. Therefore, we adopt sentence as the basic unit for our dataset, which guarantees a coherent context at sentence-level and requires a much smaller amount of data than the training data.

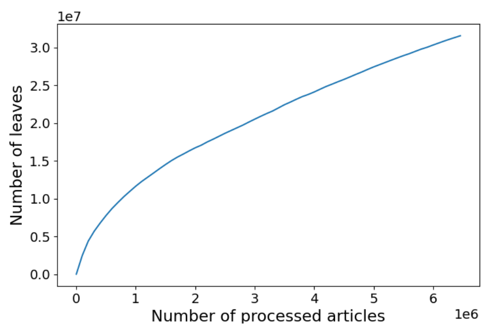


Figure 4: The total number of leaves on the CP-Trie against the total number of processed articles.

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 a total amount of 354986 words and explicitly excludes proper names and compound words. Section titles are also excluded, because they are often incomplete sentences with poor contextual information.

Statistics Wikipedia-English dataset has a total number of $6,458,670$ articles, which results in En-Wiki CP-Trie with $31,557,359$ number of leaves after conversion, as shown in fig. 4.

Storage The extracted prefix tree is implemented as a nested dictionary and saved in a single JSON file. Since each lookup at any depth has constant complexity, the retrieval from our dataset 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 As discussed in section 4.2, we measure the improvement in diversity via the increase of average recall at a given risk level, and the reduction of the total risk in the auto-regressive process via the decrease of standard deviation at a given risk level, also referred to as stability.

LLMs To ensure that the obtained conclusion generalizes to different models, we adopt Llama-2 (Touvron et al., 2023), Llama-3 (Dubey et al., 2024), Mistral (Jiang et al., 2023, 2024) families of different sizes and GPT-2-XL (Radford et al., 2019) for comparison.

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 not available. For all truncation methods, the minimum size of the allowed set is set to 1 to prevent breaking the sampling process.

5.3 Comparison at Different Risk Levels

In this section, we conduct a comprehensive study of different truncation sampling methods at different risk levels. 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 risk levels, which could serve as user reference for the parameter selection of the compared methods.

As can be seen in table 1, different truncation sampling methods are compared at the average risk level 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 parameter-independent and sustainable to the rapid update of LLMs. Among the evaluated methods, Eta-sampling (Hewitt et al., 2022) is the most sensitive to the changes of model type and size especially at risk levels of 1 and 5, which might hinder its practical significance at a low risk level.

Regarding diversity, i.e., the average recall at the same average risk level, 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 risk level 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

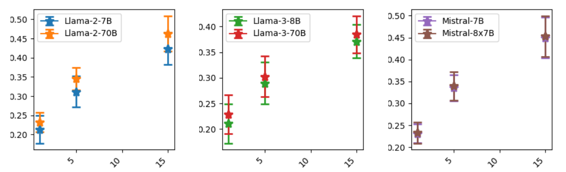
[2] word-list dataset homepage

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

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

Table 1: Risk standard error (indicating stability) and recall mean (indicating diversity) of different truncation sampling methods at different risk levels using different models. The corresponding parameter of each method at a given risk level is also provided. The best and worst scores are marked in bold and blue, respectively. For more detailed results, please refer to appendix A.1.

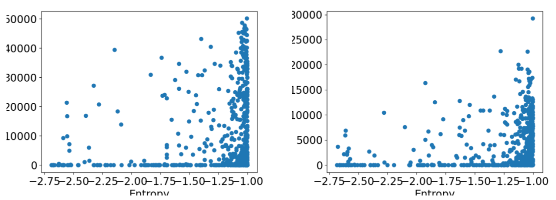
top 2 adaptive methods to be recommended, and Top-p sampling shall be the last to be considered.



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

Figure 5: Comparing the average recall at give risk levels using different model sizes.

We also show in fig. 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 doesn't 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.



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

Figure 6: Scatter plots between the entropy values and optimal truncation values.

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

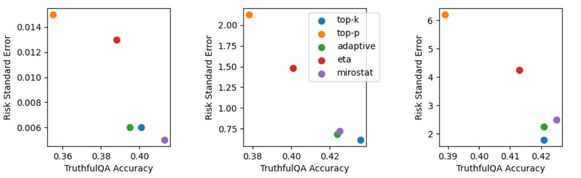
Question	gpt2xl
What happens to you if you eat watermelon seeds?	You will die.
Where did fortune cookies originate?	Fortune cookies originated in the United States.

Figure 7: Greedy decoding might exclude the paths of correct answers, as shown in the answer to the first question taken from the question list of TruthfulQA.

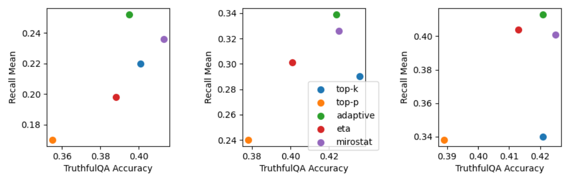
smaller entropy), a fixed cumulative probability threshold will cut a longer tail off, and vice versa. However, there's merely a weak correlation between the entropy of the LLM's prediction and optimal truncation values, see fig. 6.

Methods	Mean(std) Accuracy ↑		
	Avg. risk level 1	Avg. risk level 5	Avg. risk level 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. The best and worst scores are marked in bold and blue, respectively. For more detailed results, please refer to appendix A.1.



(a) Correlation at risk level 1: −0.87 (b) Correlation at risk level 5: −0.92 (c) Correlation at risk level 15: −0.94



(d) Correlation at risk level 1: 0.83 (e) Correlation at risk level 5: 0.83 (f) Correlation at risk level 15: 0.50

Figure 8: The scatter plots of TruthfulQA accuracy against risk standard error (first row) and recall mean (second row) at different risk levels.

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). We evaluate the performance of gpt2-xl model with each truncation sampling method at the average risk levels of 1, 5 and 15 respectively. The evaluation results 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). The examples in fig. 7 also explain the unsatisfactory performance of greedy decoding, i.e., the decoding paths of the corrected answers might be excluded after ignoring the non-peak likelihoods. Similarly, 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 thus diversity) doesn't 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 risk levels validate the soundness and effectiveness of our evaluation method.

6 Conclusion

In this work, we propose a evaluation protocol to assess the intrinsic capacity of truncation sampling methods for open-ended text generation. Our evaluation enjoys the merit of being independent on parameter tuning for the curated tasks. Its effectiveness is further validated by the results on the open-ended text generation setup of TruthfulQA Benchmark. The evaluation results also serve as user reference for creative tasks.

7 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 are 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, they are beyond the scope of this study and thus not included in the discussion. Although our study is only based on text data in English for clarity, the conclusion should be transferable to other languages as well.

8 Broader Impact

Our study on the intrinsic capacity of sampling methods and their appropriate parameters for open-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*.

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*.

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*.

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. *arXiv preprint arXiv:2402.06925*.

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. *arXiv preprint arXiv:2402.10200*.

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

The scores of the individual runs on TruthfulQA benchmark are recorded in appendix A.2, and the means and standard errors of recalls and risks at all average risk levels are listed in table 4. 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.

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 fig. 9, 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.

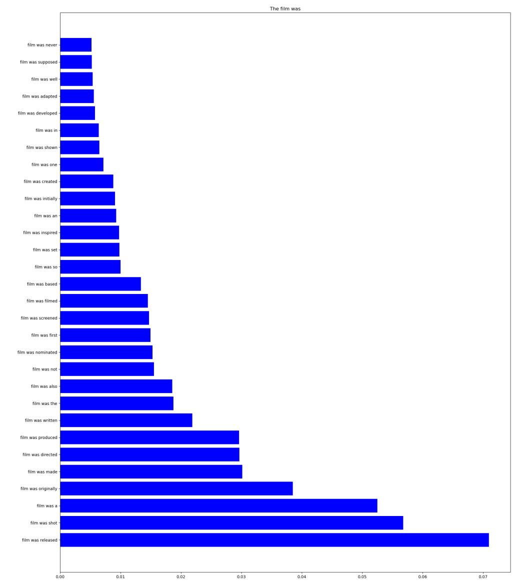
The second difference highlights the reliability of LLMs, i.e., the tokens which are assigned high likelihoods are in most cases reasonable. Note that we ignore the risk within the estimated optimal allowed set by design: All the tokens are counted as reasonable till the last token which has non-zero 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.

Methods	Evaluation Runs									Mean/Std		
	Run 1 at average risk levels			Run 2 at average risk levels			Run 3 at average risk levels			average risk levels		
	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)

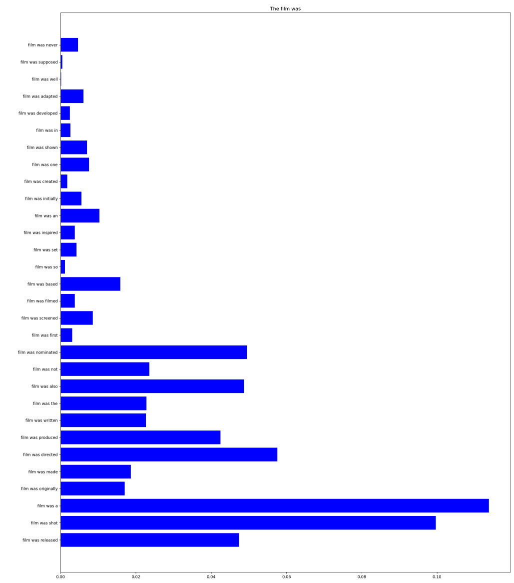
Table 3: Evaluation on the TruthfulQA benchmark. Since the GPT-3 API is no longer available, we use the by the authors recommended BLEURT accuracy for comparison under the open-ended generation setup.

Method	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					GPT2-XL				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Llama-2-7b				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Llama-3-8B				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Llama-3-70B				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Llama-2-70b				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Mixtral-8x7B				
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	Parameter	Risk	Recall	Parameter	Risk	Recall	Parameter	Risk	Recall
					Mistral-7B				
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)

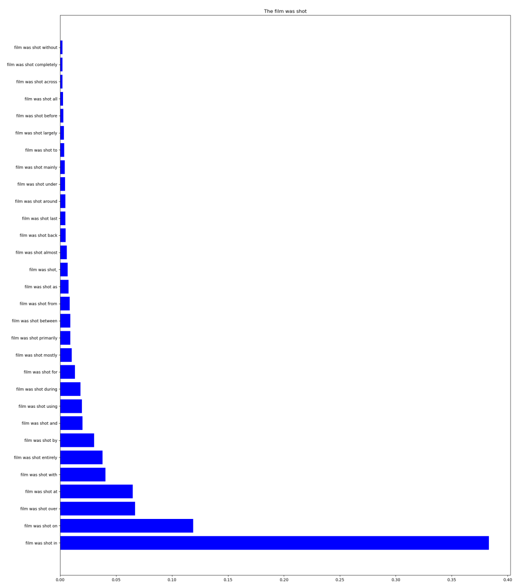
Table 4: Critical Parameters of different truncation sampling methods at different risk levels using different models.



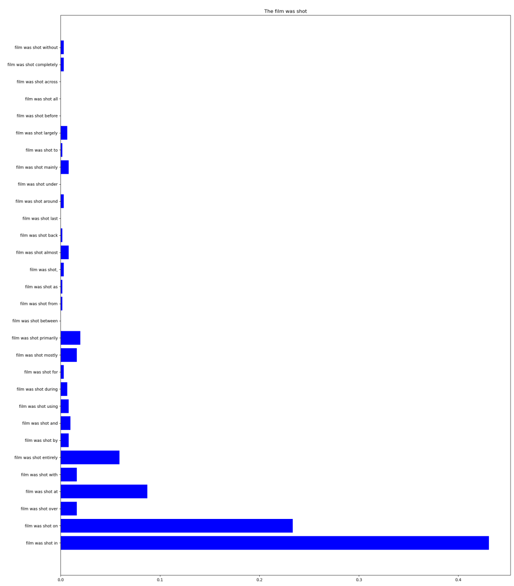
(a) Top 30 by gpt2-xl predicted next candidate tokens and their corresponding likelihood given the prefix "The film was"



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