Improving Open-Ended Text Generation via Adaptive Decoding

Wenhong Zhu [1] **Hongkun Hao** [1] **Zhiwei He** [1] **Yiming Ai** [1] **Rui Wang** [1]

Abstract

Current language models decode text token by token according to probabilistic distribution, and determining the appropriate candidates for the next token is crucial to ensure generation quality. This study introduces *adaptive decoding*, a mechanism that dynamically empowers language models to ascertain a sensible candidate set during generation. Specifically, we introduce an entropy-based metric called *confidence* and conceptualize determining the optimal candidate set as a *confidence-increasing* process. The rationality of including a token in the candidate set is assessed by leveraging the increment of confidence. Experimental results reveal that our method balances diversity and coherence well. The human evaluation shows that our method can generate human-preferred text. Additionally, our method can potentially improve the reasoning ability of language models.

1. Introduction

When it comes to open-ended text generation tasks, the aim is to produce coherent and fluent output based on limited input information (Holtzman et al., 2020). These tasks span various applications, including casual conversation (Thoppilan et al., 2022), storytelling (Mostafazadeh et al., 2016), and similar domains. Language models (LMs) presently estimate the likelihood of the next token in text generation, and they have gained extensive application in various tasks (Brown et al., 2020; Touvron et al., 2023). One factor that determines the quality of generation is the decoding algorithm.

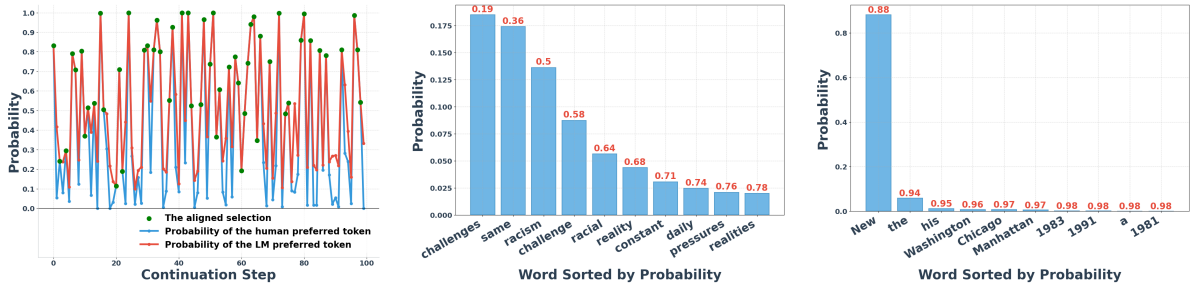
For LMs, training solely on maximum likelihood estimation (MLE) without additional fine-tuning, such as human preference alignment, often raises distinct degradation issues (Holtzman et al., 2020). These challenges are closely intertwined with decoding algorithms. The decoding algorithm can be seen as a mechanism to truncate the entire vocabulary space into the next candidate token set. When this set is limited, as seen in greedy decoding and beam search, it leads to heightened repetition as the model favors previously decoded tokens (Li et al., 2020; Zhu et al., 2023). Conversely, when this set is expansive, as in top-p sampling, it can result in incoherence and hallucination (Xu et al., 2023). This occurs because certain improbable tokens still contribute to the cumulative probability, causing issues when sampling these tokens. Numerous efforts have been devoted to addressing these challenges. Some endeavors focus on enhancing diversity within the specified sampling space (Su et al., 2022; Xu et al., 2023), while others strive to identify a rational sampling space (Meister et al., 2023b; Hewitt et al., 2022).

We provide one human-written text example for the GPT2-XL model and use teacher-forcing decoding to demonstrate the distinction between the human-preferred and LM-preferred tokens. In doing so, we save the probability of human-preferred and LM-prefered tokens. As shown in Figure 1(a), the choices made by humans and the LM predominantly align at high probability points. The distinction lies in that the portion of the token selected by humans tends to fall within the low probability region predicted by the LM. Simultaneously, the model tends to choose tokens with relatively higher probability. This suggests that the selection of the next token could have many possibilities.

On the other hand, the distribution predicted by the LM exhibits a long-tailed pattern (Holtzman et al., 2020), indicating that choosing the next token following the previous context is constrained. The main difference in the long tail distribution is the head shape. For example, as shown in Figure 1(b), the distribution is flattened, indicating that the LM has multiple choices of the next token. The diversity of generated text may be compromised if our candidate set is too limited. While in Figure 1(c), the distribution is on a sharp shape, implying the restricted choices. In this scenario, an excessively large candidate set could lead to a loss of coherence in the generated text. These observations raise the question: how do we determine the rational size of the candidate set to ensure both diversity and coherence in the generated text?

[1]MT Lab, Department of Computer Science and Engineering , Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Rui Wang <wangrui12@sjtu.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) The probability of human-preferred token and LM-preferred token.

(b) Head distribution given the **prefix**: *"Growing up, Obama faced the challenges"*

(c) Head distribution given the **prefix**: *"attended Columbia University in New"*

Figure 1. Human written text: "*Barack Obama was born in 1961. He was raised in Hawaii by his mother and grandparents. Growing up, Obama faced the challenges of being biracial, with a Kenyan father and an American mother. Despite these challenges, he excelled academically and eventually attended Columbia University in New York City.*" We provide this human-written text for GPT2-XL and use teacher-forcing decoding.

In this study, we present an *adaptive decoding* algorithm designed to dynamically determine the size of a suitable candidate set during generation. Entropy is a metric to quantify the disorder or randomness within a system. It would be useful in gauging the confidence of an LM in predicting the probability distribution of the next token. We conceptualize identifying the optimal candidate set as an *confidence-increasing* procedure. Specifically, we establish a system embracing two sets: a candidate set, initially empty, and an ordered set, initially encompassing the entire vocabulary. One metric named *confidence* was formulated to gauge system confidence based on entropy. Continuously selecting tokens with the highest probability from the ordered set and appending them to the candidate set significantly enhances the confidence of this system. The rationality of a token to join the candidate set is assessed based on the increment of confidence, allowing the model to dynamically determine a suitable set of candidates.

In our experiments, we performed two open-ended text generation tasks: document continuation and story generation. The results suggest that our approach significantly enhances diversity while preserving coherence in GPT2-XL(1.5B) and Llama2-7B models. On Llama2-7B-chat, coherence has been enhanced alongside increased diversity. Additionally, two mathematical reasoning tasks demonstrate that our method potentially improves the reasoning ability of language models. Human evaluation results show that our approach produces text that is more human-preferred.

In summary, the main contributions of this paper are as follows:

- We integrate the concept of entropy reduction into the text decoding process and design a confidence increment as the metric to assess the rationality of a token as a candidate.
- We introduce a novel *adaptive decoding* mechanism that enables LMs to determine an appropriately sized set of candidates autonomously during the generation. [1]
- *Adaptive decoding* effectively generates text in a style resembling human expression, ensuring a balance between coherence and diversity.

2. Problem Formulation

2.1. Language Models

We follow the definitions given in Hewitt et al. (2022). Let the random variable $\boldsymbol{X} = (X_1, X_2, ..., X_T)$ represents a sequence of tokens, where each X_t belongs to a finite vocabulary $\mathcal{V}$. We use $\boldsymbol{x}_{<t}$ to indicate a given prefix, x_t as a specific word in $\mathcal{V}$, and x as a random token in $\mathcal{V}$. An auto-regressive LM is a distribution $P_\theta(\boldsymbol{X})$ with parameters θ, which is factorized as $P_\theta(x) = \prod_{t=1}^{T} P_\theta(x_t|\boldsymbol{x}_{<t})$. We denote $P_\theta(X_t|\boldsymbol{x}_{<t})$ as the conditional distribution of the LM given the context $\boldsymbol{x}_{<t}$ over $\mathcal{V}$. The standard method for training an LM using maximum likelihood estimation of the training data $\mathcal{D}$, where log-likelihood is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{\boldsymbol{X} \in \mathcal{D}} \log P_{\boldsymbol{\theta}}(\boldsymbol{X}). \tag{1}$$

This is also equivalent to minimizing the forward cross-entropy between the true distribution $P^*(\boldsymbol{X})$ and $P_\theta(\boldsymbol{X})$:

$$-E_{\boldsymbol{X} \sim P^*} \sum_{t=1}^{|\boldsymbol{X}|} P^*(x|\boldsymbol{x}_{<t}) \log P_\theta(x|\boldsymbol{x}_{<t}) \tag{2}$$

And recent LMs have achieved remarkably low forward cross-entropy (Radford et al., 2019; Touvron et al., 2023).

[1] The code is available at https://github.com/zwhong714/adaptive_decoding.

Table 1. Preliminary experiment: HIT@k is a metric that measures the human preferred token falls into the top-k space of an LM predicted distribution. Results obtained by 1000 human-written text from WikiText-103.

MODEL	HIT@1	HIT@3	HIT@5
GPT2	32.37	49.89	57.01
GPT2-XL	37.09	55.21	62.46
LLAMA2-7B	47.14	65.40	72.08
LLAMA2-CHAT-7B	45.04	62.81	69.57

Recall-prioritization However, model training with MLE has inherent limitations for matching model distribution and that of human language (Meister et al., 2023b). This is essentially because cross-entropy is recall-prioritized (Meister et al., 2023a). Recall here is defined as the coverage of the model distribution P_θ over the true distribution P^*, indicating that a high recall implies tokens with high likelihood under P^* also have high likelihood under P_θ. In other words, cross-entropy focuses on increasing the model likelihood of the ground-truth next token (Ren et al., 2024). Therefore, according to Equation (1), $P_\theta(x_t|\boldsymbol{x}_{<t})$ must be nonzero for all tokens x_t in every string X in the training dataset $\mathcal{D}$, which leads to multiple possible candidates for the next token. Besides, label smoothing has become a defacto standard for large-scale training. In practice, there is more than one target next token during training due to the entropy regularization applied to the loss. As indicated in Table 1, the model prediction for the next token may not be precisely identical to the teacher-forced token. Nevertheless, it is essential to recognize that the model exhibits a significant capability in predicting the next token.

Text generation LMs can assess the likelihood of existing sequences and produce new tokens through the expression $x \sim P_\theta(\boldsymbol{X})$. The common application involves the generation of open-ended text. When presented with a sequence of m tokens extracted from natural language, denoted as $\mathcal{C} = \{x_1, ..., x_m\}$, serving as context, the LM aims to construct an n-tokens extension. This extension is decoded using the probability distribution predicted by the LM: $P_\theta(x_{m+1} : x_{m+n}|\mathcal{C}) = \prod_{t=1}^{n} P_\theta(x_t|\mathcal{C}, x_{m+1}, ..., x_{m+t-1})$. The generation of the continuation occurs token by token, employing a specific decoding strategy.

2.2. Degeneration Problems

Repetition Repetition often arises due to models trained using MLE without any intervention, leading to a gradual increase in the probability of tokens that have already occurred (Xu et al., 2022; Zhu et al., 2023). This tendency becomes particularly pronounced when utilizing greedy decoding or beam search (Holtzman et al., 2020). While this behavior aligns with the correct objective of MLE-trained models to continually maximize likelihood in the text generation process, it falls short of generating text preferred by humans. As illustrated in Table 2, this issue appears to be independent of the size of the model parameters. Even the GPT-3 base model(175B) exhibits significant repetition problems when using greedy decoding.

Table 2. Degeneration problem: repetition and incoherence

GPT3 base model (davici-002), greedy
Prefix: *"Barack Obama was born in Honolulu, Hawaii. He was born in"*
(Repetition) Continuation: *"1961. He was born on August 4, 1961. Barack Obama was born in Honolulu, Hawaii on August 4, 1961. Barack Obama was born in Honolulu"*
GPT2-XL base model, top-$p = 0.95$
Prefix: *"Barack Obama was born in Honolulu, Hawaii. He was born in"*
(Incoherence) Continuation: *"Hawaii, and he is a legitimate U.S. citizen. He's the president, and I'm the president of the United States." — Nov. 14, 2009"*

Incoherence Many decoding algorithms have been proposed to truncate low probability tokens directly and have proven to be useful (Holtzman et al., 2020; Hewitt et al., 2022; Meister et al., 2023b). They typically compute the following truncated distribution at each time step:

$$P_{\text{trunc}}\left(x \mid \boldsymbol{x}_{<t}\right) = \begin{cases} P_\theta\left(x \mid \boldsymbol{x}_{<t}\right)/Z_{\boldsymbol{x}_{<t}} & x \in \mathcal{A}_{\boldsymbol{x}_{<t}} \\ 0 & \text{o.w,} \end{cases} \quad (3)$$

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

Effective truncation is the key to improving the quality of text generation. Otherwise, there is a risk of incoherent and hallucinatory outputs, as the model may sample from low-probability tokens that may not be related to the prefix. As we can see from Table 2, the text is highly incoherent with the text generation going on when we use top-p sampling (Holtzman et al., 2020).

3. Preliminaries

Definition 3.1. Given a discrete random variable X, which takes values in the vocabulary $\mathcal{V}$ and is distributed according to $p : P_\theta(X) \to [0, 1]$, the *entropy* is defined as

$$\mathrm{H}(X) := -\sum_{x \in \mathcal{V}} p(x) \log p(x).$$

Theorem 3.2. *For any random variable X,*

$$0 \leq H(X) \leq \log |\mathcal{V}|,$$

where $|\mathcal{V}|$ denotes the size of the vocabulary $\mathcal{V}$. The upper bound is tight if and only if X is distributed uniformly on $\mathcal{V}$. The lower bound is tight if and only if X is deterministic.

By Theorem 3.2, the upper bound is tight when token distribution predicted by the LM is uniform, which means each token in the vocabulary $\mathcal{V}$ has the same probability of being sampled. However, the upper bound is often impossible to achieve in the real world. Moreover, the true upper bound is frequently linked to the model's capability to predict the next token.

To mitigate the impact of the upper bound, we normalized entropy using the min-max approach. (1) This ensures that all changes fall within a similar range, enabling the comparison and analysis of distributions predicted by different models. (2) It also facilitates comparing and analyzing entropy reduction processes.

Definition 3.3. For any random variable X, the *confidence* $\text{Conf}(X)$ is defined as the min-max scaling of entropy, and the range is $[0, 1]$:

$$\text{Conf}(X) := 1 + \frac{\sum_{x \in \mathcal{V}} p(x) \log p(x)}{\log |\mathcal{V}|}.$$

The expression $\sum_{x \in \mathcal{V}} p(x) \log p(x)$ can be conceptually divided into two components. The first component is the known part $\sum_i^k p_i \log p_i$, where k represents the number of known items. The second component is the unknown part, introducing maximum uncertainty, expressed as $(1 - \sum_i^k p_i) \log \frac{1-\sum_i^k p_i}{|\mathcal{V}|-k}$. Consequently, we can define a pivot k to divide the distribution $P_\theta(X)$ based on the number of known items. This metric related to k is as follows:

$$\begin{aligned} \text{Conf}_k(X) = 1 + \frac{1}{\log |\mathcal{V}|} (& \sum_i^k p_i \log p_i \\ & + (1 - \sum_i^k p_i) \log \frac{1 - \sum_i^k p_i}{|\mathcal{V}| - k}), \end{aligned} \tag{4}$$

where p_i is the abbreviation of $p(x_i)$.

3.1. $\text{Conf}_k(X)$ Is Monotonically Increasing

We define the rationale for including a token in the candidate set as the extent to which the presence or absence of this token affects the model's confidence. Specifically, it is defined as the difference in Equation (4) between neighboring pivots.

$$\begin{aligned} \Delta\text{Conf} &= \frac{1}{\log |\mathcal{V}|} (\text{Conf}_k(X) - \text{Conf}_{k-1}(X)) \\ &= \frac{1}{\log |\mathcal{V}|} (p_k \log p_k + (1 - \sum_i^k p_i) \log \frac{(1 - \sum_i^k p_i)}{|\mathcal{V}| - k} \\ &- (1 - \sum_i^{k-1} p_i) \log \frac{(1 - \sum_i^{k-1} p_i)}{|\mathcal{V}| - k + 1}). \end{aligned} \tag{5}$$

In Appendix C, we demonstrate that Equation (5) is greater than zero when $P_\theta(X)$ is sorted in decreasing order. Under this condition, confidence steadily increases as k continues to grow.

3.2. Descending Trend in ΔConf

Although ΔConf may lack a clear monotonic property, assessing its variability is feasible by examining its upper and lower bounds. Utilizing Corollary C.2, we can establish the lower bound as $\Delta\text{Conf} \geq \frac{p_k}{\log |\mathcal{V}|} \log(|\mathcal{V}| - k + 1)$, while employing both Corollary C.1 and Corollary C.2 allows us to derive the upper bound, namely $\Delta\text{Conf} \leq \frac{(1-\sum_i^{k-1} p_i)}{\log |\mathcal{V}|} \log(|\mathcal{V}| - k + 1)$.

As the number of known items k increases, it is observed that the probability p_k decreases. Additionally, both $\log(|\mathcal{V}| - k + 1)$ and $(1 - \sum_i^{k-1} p_i)$ exhibit a decreasing trend. Ultimately, the upper and lower bounds converge to zero. This implies that ΔConf also demonstrates a decreasing trend, a pattern influenced by the interplay between the probability p_k and the number of known items k.

3.3. Approximation by Truncation

As illustrated in Figure 2, the token distribution predicted by the LM exhibits a pronounced long-tailed pattern. Notably, this phenomenon becomes more conspicuous as the LM parameters increase and a human-alignment operation follows.

Corollary 3.4. *When a distribution exhibits a long-tail shape, the entropy of the truncated distribution $\text{H}(X(k))$ is approximately equal to the original entropy and an error within a smaller range of ϵ.*

$$\text{H}(X) = \text{H}(X(k)) + \epsilon, \tag{6}$$

where $X(k)$ follows the truncated distribution as defined by the Equation (3).

Equation (6) shows we can determine a reasonable candidate set size k by setting a reasonable threshold ϵ.

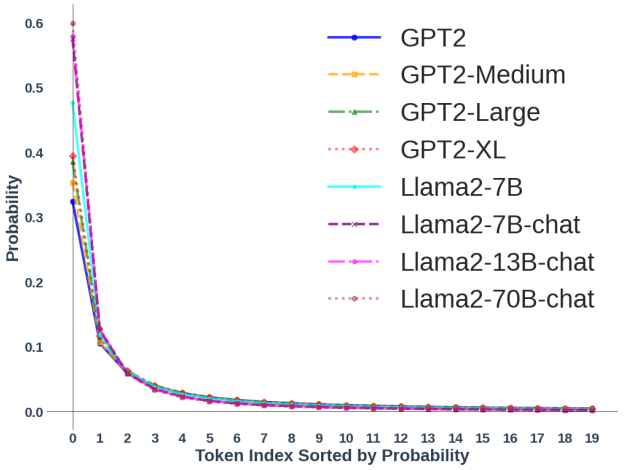


Figure 2. Preliminary experiment: Averaged top-20 token distribution of different LMs on 1000 human-written texts from WikitText-103.

4. Adaptive Decoding

Our method adheres that the optimal threshold must balance diversity (i.e., including as many tokens as possible in the candidate set) and coherence (i.e., avoiding the inclusion of tokens outside the true support) (Finlayson et al., 2024).

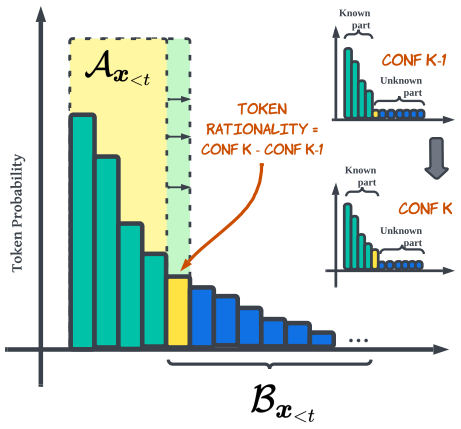


Figure 3. How adaptive decoding works. Expanding the candidate set $\mathcal{A}_{\boldsymbol{x}_{<t}}$ depends on the rationality of the token. This rationality is determined by the extent to which the model's confidence increases when considering the inclusion of this token.

Initialization For clarity, we define the known portion, as discussed in Section 3, as the next token candidate set $\mathcal{A}_{\boldsymbol{x}_{<t}}$, and the unknown part as $\mathcal{B}_{\boldsymbol{x}_{<t}}$. The initialization involves setting $\mathcal{A}_{\boldsymbol{x}_{<t}}$ to an empty set and $\mathcal{B}_{\boldsymbol{x}_{<t}}$ to an ordered collection of the entire vocabulary. The order is based on the descending probabilities predicted by the LM.

Truncation As illustrated in Figure 3, identifying a suitable set of candidates can be viewed as an iterative procedure. This involves transferring the token with the highest probability from $\mathcal{B}_{\boldsymbol{x}_{<t}}$ to $\mathcal{A}_{\boldsymbol{x}_{<t}}$. Subsequently, we calculate the incremental change in the Conf metric, as defined in Equation (5). This increment serves as a measure of the rationality of a token within the candidate set. The iteration continues until the rationality falls below a pre-set threshold, denoted as ϵ. At this point, we finalize $\mathcal{A}_{\boldsymbol{x}_{<t}}$ as our truncation space.

Computational complexity Efficient implementation of adaptive decoding is achievable through Algorithm 1. Firstly, sorting words by their predicted probabilities can be accomplished in $\mathcal{O}(|\mathcal{V}|\log|\mathcal{V}|)$. Secondly, the calculation of the increment ΔConf, as described in Equation (5), is a $\mathcal{O}(|\mathcal{V}|)$ operation. By getting ΔConf, we can effectively identify the last index with a value below the pre-set threshold in $\mathcal{O}(\log|\mathcal{V}|)$. In summary, our algorithm exhibits a time complexity of $\mathcal{O}(|\mathcal{V}|\log|\mathcal{V}|)$. However, we can fully leverage the parallel computing power of GPUs to accelerate the computation process.

Algorithm 1 Adaptive Decoding Algorithm

Input: Prefix $\mathcal{C} = \{x_1, \ldots, x_m\}$; Language Model P_θ; Vocabulary size $|\mathcal{V}|$; Threshold ϵ
Output: Continuation $\mathcal{G} = \{x_{m+1}, x_{m+2}, \ldots, x_{m+n}\}$
$\mathcal{G} \leftarrow \{\}$
for $t = m+1$ **to** $m+n$ **do**
  $p \leftarrow P_\theta(x_t|\mathcal{C})$
  $p \leftarrow \text{sort}(p, \text{decreasing order})$
  cumsum $\leftarrow$ cumulative sum of p
  residual_index $\leftarrow |\mathcal{V}| - \text{range}(1, |\mathcal{V}|+1)$
  $term1 \leftarrow p \cdot \log\left(\frac{p \cdot \text{residual\_index}}{1-\text{cumsum}}\right)$
  $term2 \leftarrow \log\left(\frac{1-\text{cumsum}}{\text{residual\_index}}\right) - \log\left(\frac{1-\text{cumsum}+p}{\text{residual\_index}+1}\right)$
  $\Delta\text{Conf} \leftarrow \frac{term1+(1-\text{cumsum}+p)\cdot term2}{\log|\mathcal{V}|}$
  $k \leftarrow \max(1, \text{LastIndex}(\Delta\text{Conf} > \epsilon))$
  ▷ top-k sampling
  $x_t \leftarrow$ sample from $\mathcal{V}^{(k)}$
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{x_t\}$
end for
Return: $\mathcal{G}$

5. Experiments

5.1. Setups

Models We assess the performance of our method on the GPT2-XL model (Radford et al., 2019), the Llama2-7B

Table 3. Main experiments are conducted by repeating the sampling process five times for each prefix, and the average score is reported. Metrics where the algorithm outperforms others are highlighted in bold. * denotes a significant difference compared to other methods ($p < 0.05$).

LM	Decoding	WritingPrompts						WikiText-103					
		rep-2	**rep-3**	**rep-4**	**Diversity**	**MAUVE**	**Coherence**	**rep-2**	**rep-3**	**rep-4**	**Diversity**	**MAUVE**	**Coherence**
	Human	3.63	0.87	0.39	95.27	–	27.12	5.42	1.20	0.41	93.18	–	62.94
GPT2-XL	greedy	82.00	80.62	79.66	1.82	0.61	**56.27***	71.86	68.33	65.95	6.98	2.03	57.81
	top-k	12.78	6.68	4.70	80.50	22.28	47.69	9.96	3.57	1.96	86.08	77.15	**61.19**
	mirostat	58.48	51.81	47.32	18.17	5.61	50.10	45.38	36.21	30.74	31.11	36.06	60.51
	top-p	12.72	6.86	4.96	80.24	21.20	48.01	8.11	2.74	1.56	88.82	78.22	60.00
	typical	13.28	7.62	5.83	79.51	20.36	46.98	6.73	2.13	1.19	90.71	78.49	59.66
	η-sampling	8.92	4.43	3.33	86.26	22.12	45.15	7.80	2.55	1.45	89.44	77.94	59.90
	adaptive (ours)	**6.86***	**3.66***	**3.07***	**89.30***	**25.18***	37.71	**5.20***	**1.43***	**0.80***	**93.23***	**79.57**	55.98
	Human	7.44	1.59	0.64	90.67	–	32.84	4.73	1.00	0.33	94.10	–	60.34
Llama2-7B	greedy	77.32	74.67	72.84	5.58	1.51	46.40	43.71	36.70	32.09	34.54	22.28	55.85
	top-k	9.13	4.45	3.27	87.18	23.57	23.02	7.18	2.44	1.20	90.01	74.80	58.49
	top-p	8.33	4.10	3.29	88.38	23.60	23.05	7.07	2.49	1.33	90.05	77.19	58.19
	mirostat	13.93	8.49	6.65	79.71	21.39	**56.23***	12.45	6.14	3.92	80.63	69.22	**65.79***
	typical	7.16	3.55	2.91	90.33	25.78	22.56	7.06	2.48	1.27	90.08	76.04	57.96
	η-sampling	**6.60***	2.96	2.35	**90.99**	24.43	23.19	**5.93***	**1.84***	**0.90***	**91.96***	**78.26**	57.46
	adaptive (ours)	7.02	**2.63***	**1.58***	89.98	**35.26***	35.10	6.81	2.31	1.24	90.55	76.74	58.22

model, and its associated chat version (Touvron et al., 2023).

Datasets We explore two open-ended text generation applications: document continuation using the WikiText-103 dataset (Merity et al., 2017), which contains a large collection of Wikipedia articles. Another is story generation on the WritingPrompts dataset (Fan et al., 2018), a notably challenging endeavor. This task involves crafting imaginative continuations based on abstract, high-level story prompts contributed by online users.

Baselines Our proposed algorithm is compared against various decoding algorithms. Top-k method (Fan et al., 2018) maximizes $\sum_{x\in\mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta\left(x \mid \boldsymbol{x}_{<t}\right)$ while constraining $|\mathcal{A}_{\boldsymbol{x}_{<t}}| = k$. Here, k serves as the predefined hyperparameter. Top-p method (Holtzman et al., 2020) minimizes $|\mathcal{A}_{\boldsymbol{x}_{<t}}|$ while constraining $\sum_{x\in\mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta\left(x \mid \boldsymbol{x}_{<t}\right) \geq p$. p is the preset hyperparameter. Mirostat method (Basu et al., 2021) is perplexity-controlled by estimating $|\mathcal{A}_{\boldsymbol{x}_{<t}}|$ based on Zipf's law and the target cross-entropy. Typical decoding (Meister et al., 2023b) tries to minimize the absolute distance $\sum_{x\in\mathcal{A}_{\boldsymbol{x}_{<t}}} \mid \mathrm{H}\left(X_t \mid \boldsymbol{X}_{<t}=\boldsymbol{x}_{<t}\right)+\log P_\theta\left(x \mid \boldsymbol{x}_{<t}\right) \mid$, by constraining the $\sum_{x\in\mathcal{A}_{\boldsymbol{x}_{<t}}} P_\theta\left(x \mid \boldsymbol{x}_{<t}\right) \geq \tau$. τ is the preset hyperparameter. η-sampling (Hewitt et al., 2022) considers the allowed set $\mathcal{A}_{\boldsymbol{x}_{<t}} = \{x \in \mathcal{V} | P_\theta(x|\boldsymbol{x}_{<t}) > \eta\}$, where $\eta = \min\left(\epsilon, \alpha \exp \mathrm{H}\left(X_t \mid \boldsymbol{X}_{<t}=\boldsymbol{x}_{<t}\right)\right)\}$, and α and ϵ are hyperparameters.

Automatic evaluation **Repetition** is defined as **rep-n** $= 1.0 - \frac{\#\text{unique n-gram}}{\#\text{total n-gram}}$ to measure sequence-level repetition according to the portion of duplicate n-grams (Welleck et al., 2020). **Diversity** considers the repetition of generated text at different n-gram levels and can be calculated as follows: $\prod_{n=2}^{4}(1.0 - \frac{\textbf{rep-n}}{100})$ (Meister et al., 2023b). **MAUVE** (Pillutla et al., 2021) is a metric that quantifies the similarity in token distribution between generated text and human-written text. Semantic **coherence** is defined as $v_{\mathcal{C}}^{\top} v_{\mathcal{G}} / \left(|v_{\mathcal{C}}| \cdot |v_{\mathcal{G}}|\right)$, where $v_{\mathcal{C}} = \mathrm{SimCSE}(\mathcal{C})$ and $v_{\mathcal{G}} = \mathrm{SimCSE}(\mathcal{G})$. It is computed as the cosine similarity between the sentence embeddings of the prefix and the continuation, represented by SimCSE (Gao et al., 2021).

5.2. Implementation Details

We randomly select 1200 data samples from the training set of each dataset, use 1000 samples of them to evaluate different decoding algorithms and use the remaining 200 samples to select hyperparameters. Hyperparameter scans can be found in Appendix D, and we select the hyperparameters that result in the optimal MAUVE score (Meister et al., 2023b) to guarantee fairness. In the context of WritingPrompts, we directly employ the original prompt from the dataset to feed models to execute the content continuation task. In the case of WikiText-103, a prefix with fixed-length 32 is utilized for the continuation task. The maximum generation length is constrained to 256 tokens for both datasets. To ensure robustness, we conduct the sampling process five times for each prefix and subsequently report the average score. Furthermore, all decoding methods adopt multinomial sampling within the truncated space.

5.3. Results

The primary experiments are presented in Table 3. The MAUVE score is measured by referencing human-written texts. From the perspective of human texts, there is a high diversity. Since the WritingPrompts can have many story continuations given a prompt, coherence on this dataset is low even when written by humans. It can be seen that using the greedy decoding method for LMs trained by MLE would produce a serious repetition phenomenon, leading to high coherence but low MAUVE metric and low diversity. This indicates that the generated text is in contrast to the human-written text. Conversely, this phenomenon is significantly improved when we expand the set of allowable samples for the next token. Therefore, we must combine these metrics to analyze the results in Table 3.

When utilizing the adaptive decoding method, the two LMs generally perform better than other methods on two datasets regarding MAUVE and diversity. Besides, on the WritingPrompts dataset, we improved coherence compared with written text by humans. This observation suggests that our approach is well-suited for datasets like WritingPrompts, which excels in producing higher-quality text with limited prompts. In contrast to our approach, Mirostat determined the size of candidate sets by managing perplexity, resulting in a notable enhancement in coherence. Nevertheless, upon examining the diversity and MAUVE indicators, it appears this could be attributed to partial repetition of generation. Additionally, we found that η-sampling is quite competitive to our proposed method, and we perform a detailed comparison in Section 6.

5.4. Human Evaluation

We randomly select 200 samples from WikiText-103 for human evaluation, as the prefixes of this dataset offer ample contextual information. The evaluation guidance is outlined in Appendix F, with the results presented in Table 4.

Table 4. Human evaluation of 200 WikiText-103 samples using the Llama2-7B model. * denotes a significant difference compared to other methods ($p < 0.05$).

A Is Better		Neutral	B Is Better	
adaptive	4.000	45.000	**51.000***	human
adaptive	**64.500***	17.000	18.500	top-k
adaptive	**64.000***	15.000	21.000	top-p
adaptive	**50.125***	30.875	19.000	η-sampling

Observing Table 4, it becomes evident that while the adaptive decoding algorithm falls short of achieving parity with human-generated content, a notable 45% of the evaluated data poses a challenge in distinguishing between human-written and adaptively decoded text. Simultaneously, it is apparent that text decoded through adaptive decoding surpasses that generated through top-k, top-p, and η-sampling algorithms. Furthermore, during human evaluation, we found that η-sampling occasionally generates excessively short text.

5.5. Adaptive Decoding for Policy Fine-tuned Model

Llama 2-chat undergoes fine-tuning with an objective aligned to human preferences rather than MLE, as detailed in Touvron et al. (2023). We evaluate the performance of adaptive decoding compared to top-p sampling on the Llama2-7B-chat model. We utilized its official generation configuration with the temperature set to 0.6. As illustrated in Table 5, our decoding algorithm demonstrates the ability to enhance generation quality. This improvement is marked by increased diversity without compromising coherence.

Table 5. Experiments are conducted by repeating the sampling process five times for each prefix from WritingPrompts, and the average score is reported. p is set to 0.95 while threshold for adaptive decoding is set to 0.0005. * denotes a significant difference compared to other methods ($p < 0.05$).

Decoding	Llama2-7B-chat		
	MAUVE	Diversity	Coherence
top-p	31.85	87.41	49.85
adaptive	**37.06***	**93.99***	**54.39***

Besides, we demonstrate the usefulness of our method on real-world test sets such as the MT-Bench benchmark (Zheng et al., 2024). We utilize GPT-4 (OpenAI et al., 2023) to assess the generations produced by top-p and adaptive decoding methods, maintaining the same evaluation settings as Zheng et al. (2024). This approach demonstrates a notable agreement rate exceeding 80% between human judges and GPT-4 evaluation. Regarding MT-bench experiments for LLama2-7B-chat testing, the results are summarized in Table 6:

Table 6. Evaluating text generation on the MT-Bench dataset using GPT-4 (score: $0 \sim 10$). p is set to 0.95 while the threshold for adaptive decoding is set to 0.0005.

Decoding	Llama2-7B-chat		
	turn-1	turn-2	mean
top-p	6.7688	5.7625	6.2656
adaptive	**6.9000**	**5.9750**	**6.4375**

5.6. Reasoning Tasks

We utilize Llama2-7B-chat and Llama2-13B-chat models to tackle mathematical reasoning tasks in GSM8K (Cobbe et al., 2021) and MultiArith. We randomly selected 300 samples from each dataset and instructed the models with

the same instruction shown in Appendix E. The results were checked by humans.

Table 7. Zero-shot evaluation. p is set to 0.95 while threshold for adaptive decoding is set to 0.0005. All the models use their chat version.

Decoding	GSM8K		
	Llama2-7B	**Llama2-13B**	**Llama2-70B**
greedy	29.33	39.33	56.33
top-p	29.00	37.00	56.00
adaptive	**30.00**	**40.33**	**58.67**
Decoding	**MultiArith**		
	Llama2-7B	**Llama2-13B**	**Llama2-70B**
greedy	69.33	66.67	84.67
top-p	68.33	**73.33**	85.33
adaptive	**69.33**	71.33	**87.00**

Initially, we observed that greedy decoding yielded satisfactory results, as these models were fine-tuned based on human instruction, establishing our baseline. Subsequently, we expanded the candidate size to investigate whether the models exhibited sycophancy or encountered other issues with reasoning, potentially leading to decreased performance. It was observed that increasing the candidate size improved performance for relatively smaller models, and the effect was the opposite for larger models. Nonetheless, our adaptive decoding strategy maintained performance levels.

6. Analysis

6.1. Ablation Study

Effect of threshold As outlined in Section 3.2, ΔConf strongly correlates with the current probability and the size of the allowed set. Consequently, adjusting various thresholds empowers us to manage both of them. The results in Table 8 show that the hyperparameter ϵ is readily controllable. As ϵ increases, the allowed set size k diminishes, maintaining a relatively constant cumulative probability within the set while exhibiting an increase in standard deviation. Concurrently, there is an observable rise in model confidence. This trend suggests heightened confidence levels may lead to degradation issues within a constrained allowable set.

Effect of sampling ways Beyond the significance of truncation space, the method employed for sampling within this space plays a crucial role. In our investigation, we compare random sampling with multinomial sampling. The latter method preserves the probability distribution of the allowed set and samples tokens according to that distribution. As illustrated in Table 8, when considering the GPT2-XL model, different sampling methods exhibit only marginal impacts on the results. This can be attributed to the observation that, for the smaller LM, the distribution within the top-k space tends to flatten, as depicted in Figure 2.

Table 8. Effect of threshold and sampling ways. Results are obtained from WritingPrompts.

LM	ϵ	**Random**		
		k	**cumulative prob.**	**Conf**
GPT2-XL	0.001	30.34 ± 13.23	77.34 ± 14.67	64.89 ± 14.13
	0.005	8.13 ± 4.17	73.20 ± 18.40	71.82 ± 14.12
	0.01	4.31 ± 2.58	70.44 ± 21.36	75.30 ± 14.83
	0.02	1.80 ± 1.27	75.28 ± 26.22	84.06 ± 16.30
Llama2	0.001	31.81 ± 15.23	79.22 ± 13.84	65.08 ± 14.91
	0.005	8.14 ± 4.67	77.65 ± 17.07	74.41 ± 14.41
	0.01	4.51 ± 2.70	73.90 ± 19.78	77.00 ± 14.05
	0.02	2.13 ± 1.47	75.51 ± 23.15	83.89 ± 14.21
LM	ϵ	**MultiNomial**		
		k	**cumulative prob.**	**Conf**
GPT2-XL	0.001	30.40 ± 12.99	77.50 ± 14.59	64.89 ± 13.92
	0.005	8.21 ± 4.17	73.20 ± 18.35	71.77 ± 14.03
	0.01	4.25 ± 2.57	70.17 ± 21.75	75.20 ± 15.12
	0.02	1.73 ± 1.23	76.93 ± 25.70	85.15 ± 15.98
Llama2	0.001	21.73 ± 15.79	89.09 ± 11.00	77.17 ± 15.12
	0.005	6.47 ± 4.56	83.26 ± 16.32	80.44 ± 14.53
	0.01	3.39 ± 2.61	81.39 ± 19.45	83.76 ± 14.61
	0.02	1.67 ± 1.27	83.00 ± 22.63	89.00 ± 14.05

However, in the case of the Llama2-7B model, an interesting observation emerges: as ϵ increases, the multinomial method enhances model confidence and converges on a restricted allowed set. This phenomenon may arise from multinomial sampling being driven by the probability distribution, leading the model to favor tokens with the highest probabilities. Consequently, this contributes to degradation issues for models trained with MLE. Conversely, based on the outcomes of random sampling, including a few low-probability tokens appears to mitigate this degradation phenomenon. This suggests we need to lower ϵ when using multinomial sampling.

6.2. Deocding Latency

We employed these decoding algorithms to generate text based on 100 distinct prefixes. Subsequently, we computed the average token decoding time (after deriving logit from the model). Our implementation relies on the top-p algorithm. As observed from Table 9, there is a clear trend of decreasing average token decoding time with a reduced vocabulary size.

6.3. Qualitative Analysis

We compare the truncation behaviors of different decoding algorithms through examples extracted from Ribeiro et al. (2020).

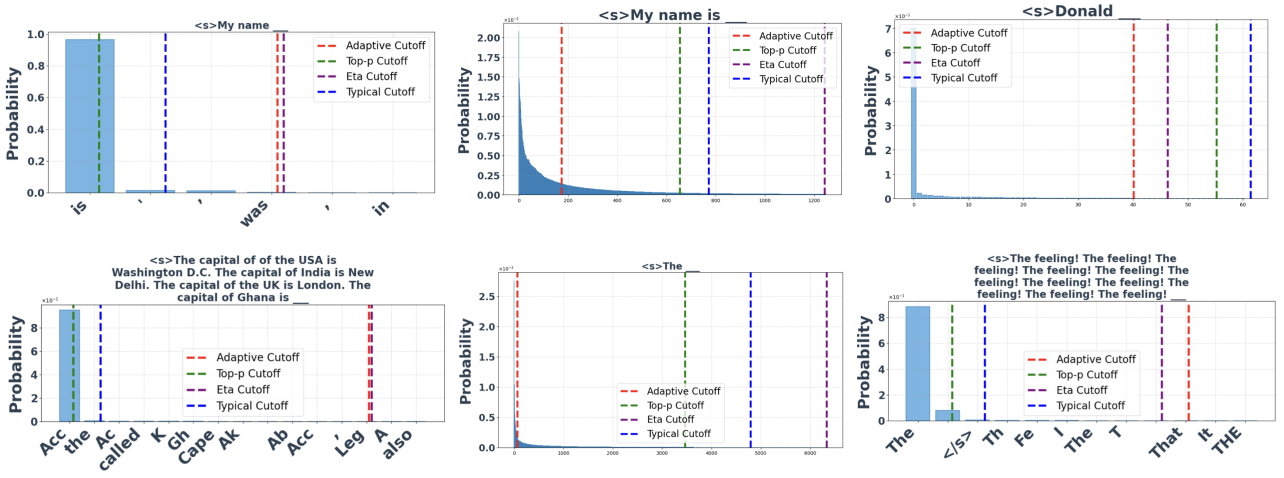


Figure 4. Unit tests of the truncation behavior of top-p, typical, η-sampling and adaptive decoding on CheckList-inspired prefixes. $p = 0.90$, typical$= 0.90$, $\eta = \sqrt{0.0009} = 0.03$, and threshold for adaptive decoding ϵ is 0.0005.

Table 9. Decoding latency. The computational hardware is NVIDIA RTX 3090, with the model loaded in float16. The unit of measurement is milliseconds (ms).

Decoding	GPT2-XL	Llama2-7B
greedy	0.0221	0.0265
top-k	0.3116	0.3466
top-p	0.4085	0.3706
adaptive	0.5741	0.5265

In our analysis of the top-p algorithm, we observed that both top-p and typical decoding exhibit similar truncation behaviors. They effectively truncate low-entropy distributions but are less suitable for high-entropy distributions. Adaptive decoding shows behavior similar to η-sampling for low-entropy distributions, offering enhanced diversity compared to the previous two methods. However, our method uniquely establishes a bounded candidate set for high-entropy distributions, setting it apart from the other three approaches.

7. Related Work

Deterministic algorithms Establishing a reasonable size for the next candidate token set seems challenging. Greedy decoding directly uses the highest probability as the next token. Beam search maintains multiple probable tokens as the candidates of the next tokens. However, they often lead to high repetition. Contrastive search (Su et al., 2022) predefines the top-k space and selects the discriminative token concerning the previous context. However, it is time-consuming and may select the irrational token. Penalty decoding aims to improve greedy decoding by adding penalties to tokens that have appeared before since the model tends to increase the probability of these tokens (Zhu et al., 2023). Look-back decoding (Xu et al., 2023) adds randomness based on greedy decoding, which uses KL divergence to track the distribution distance between the current and historical decoding steps and limit the next token probability distribution to a reasonable distance from the history.

Stochastic algorithms The most typical random decoding algorithms are top-k (Fan et al., 2018) and top-p (Holtzman et al., 2020)algorithms. Some later works are proposed to improve the truncation space. Mirostat (Basu et al., 2021) also proposes an adaptive decoding algorithm. In contrast to our work, they aim to control the perplexity close to a target value. They first calculate one value assuming words follow Zipf's law and then design a function with this value to calculate the size of the allowed set. η-sampling (Hewitt et al., 2022) proposes to truncate words below an entropy-dependent probability threshold. Typical decoding (Meister et al., 2023b) measures the distance between negative log probabilities and conditional entropy and then takes words from this list to cover p percent of the distribution. Recently, Finlayson et al. (2024) proposed basis-aware-threshold (BAT) sampling that uses the softmax matrix to find tokens that might have non-zero true probability without relying on a threshold.

8. Conclusion

In conclusion, our research introduces a novel decoding algorithm named *adaptive decoding* for open-ended text generation, leveraging the thought of entropy reduction to dynamically determine the size of a suitable candidate set. We demonstrated its effectiveness in balancing coherence and diversity through experiments, outperforming existing decoding methods across different language model scales.

Impact Statement

This paper presents work that aims to advance the field of natural language generation. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (62176153) and the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102, as the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University).

References

Basu, S., Ramachandran, G. S., Keskar, N. S., and Varshney, L. R. Mirostat: a neural text decoding algorithm that directly controls perplexity. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082.

Finlayson, M., Hewitt, J., Koller, A., Swayamdipta, S., and Sabharwal, A. Closing the curious case of neural text degeneration. In *The Twelfth International Conference on Learning Representations*, 2024.

Gao, T., Yao, X., and Chen, D. SimCSE: Simple contrastive learning of sentence embeddings. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552.

Hewitt, J., Manning, C., and Liang, P. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Li, M., Roller, S., Kulikov, I., Welleck, S., Boureau, Y.-L., Cho, K., and Weston, J. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4715–4728, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.428.

Meister, C., Pimentel, T., Malagutti, L., Wilcox, E., and Cotterell, R. On the efficacy of sampling adapters. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1437–1455, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.80.

Meister, C., Pimentel, T., Wiher, G., and Cotterell, R. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023b.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098.

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman,

A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. Gpt-4 technical report, 2023.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4816–4828. Curran Associates, Inc., 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ren, S., Wu, Z., and Zhu, K. Q. EMO: EARTH MOVER DISTANCE OPTIMIZATION FOR AUTO-REGRESSIVE LANGUAGE MODELING. In *The Twelfth International Conference on Learning Representations*, 2024.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442.

Su, Y., Lan, T., Wang, Y., Yogatama, D., Kong, L., and Collier, N. A contrastive framework for neural text generation. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 21548–21561. Curran Associates, Inc., 2022.

Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhao, V., Zhou, Y., Chang, C.-C., Krivokon, I., Rusch, W., Pickett, M., Srinivasan, P., Man, L., Meier-Hellstern, K., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Aguera-Arcas, B., Cui, C., Croak, M., Chi, E., and Le, Q. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez,

V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020.

Xu, J., Liu, X., Yan, J., Cai, D., Li, H., and Li, J. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3082–3095. Curran Associates, Inc., 2022.

Xu, N., Zhou, C., Celikyilmaz, A., and Ma, X. Look-back decoding for open-ended text generation. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1039–1050, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.66.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

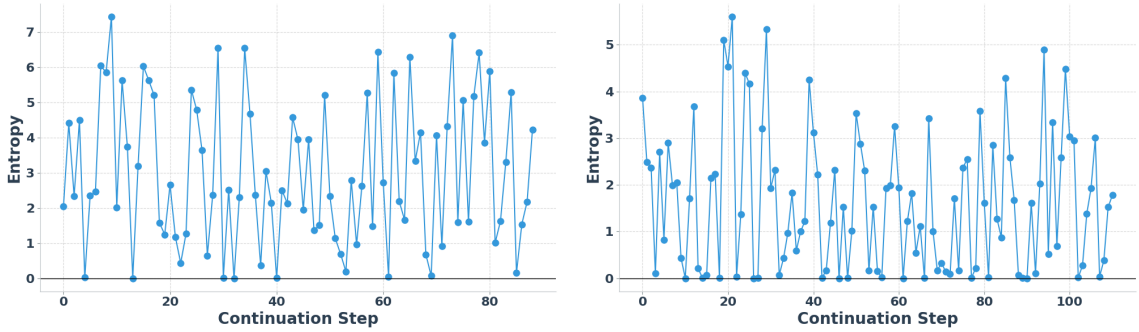
Zhu, W., Hao, H., and Wang, R. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1218–1228, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.78.

A. Limitations

We acknowledge that implementing the adaptive decoding algorithm introduces additional overhead. We hope that future work can address and eliminate this overhead. While we have theoretically demonstrated the rationale behind our method, further experimental validation through more extensive reasoning tasks is needed to confirm its effectiveness. Additionally, we anticipate the development of improved metrics for evaluating the results of model generation. Moreover, our verification of the decoding algorithm has been limited to the language models only and has not yet been extended to other areas like the multimodal domain.

B. Entropy Change

It is evident from Figure 5 that the entropy of the model experiences notable fluctuations during text generation. Furthermore, distinct word segmentation among various models yields varying upper bounds for text generation entropy in each model. The amplitude of entropy changes between models is also influenced by the quality of model pre-training. Therefore, normalizing entropy becomes important to mitigate the impact of dimensional effects.



(a) The value of entropy at each generation step of the GPT2-XL model guided by human-written text.

(b) The value of entropy at each generation step of the Llama2-7B model guided by human-written text.

Figure 5. Human written text: "Barack Obama was born in 1961. He was raised in Hawaii by his mother and grandparents. Growing up, Obama faced the challenges of being biracial, with a Kenyan father and an American mother. Despite these challenges, he excelled academically and eventually attended Columbia University in New York City."

C. Proof

C.1. Proof of Section 3.1

Examine Equation (5) closely, and it becomes evident that $p_k \log p_k$ represents the average uncertainty carried into the known segment. On the other hand, the latter term $(1 - \sum_i^k p_i) \log \frac{(1-\sum_i^k p_i)}{|\mathcal{V}|-k} - (1 - \sum_i^{k-1} p_i) \log \frac{(1-\sum_i^{k-1} p_i)}{|\mathcal{V}|-k+1}$ reflects the certainty introduced into the unknown portion. It can be shown that the increase in certainty surpasses the introduction of uncertainty as the known part expands.

Corollary C.1. *When sorting the state distribution X predicted by the language model in descending order after applying the softmax function, it follows that $p_k \geq \frac{1-\sum_i^{k-1} p_i}{|\mathcal{V}|-k+1}$.*

Proof. We acknowledge that $p_1 > p_2 > ... > p_{|\mathcal{V}|}$ with $\sum_i^{|\mathcal{V}|} p_i = 1$. If, however, $p_k < \frac{1-\sum_i^{k-1} p_i}{|\mathcal{V}|-k+1}$, it implies the existence of j within the range $[k+1, |\mathcal{V}|]$ such that $p_j > p_k$. This assertion results in a contradiction. □

Corollary C.2. *Given that the sum of probabilities for all tokens in the vocabulary is expressed as $\sum_i^{|\mathcal{V}|} p_i = 1$, we can deduce that $1 - \sum_i^{k-1} p_i - p_k = \sum_{k+1}^{|\mathcal{V}|} p_i \geq 0$. Observing this inequality, it is evident that $1 - \sum_i^{k-1} p_i \geq p_k$.*

Corollary C.3. *The function $Conf_k(X)$ is monotonically increasing, as it is always guaranteed that $\Delta Conf \geq 0$.*

C.2. Proof of Section 3.3

Proof. It is intuitively correct that $\mathrm{H}(X) = -\sum_i^{\mathcal{V}} p_i \log p_i$ converges to $-\sum_i^k p_i \log p_i$ under the condition $P_\theta(X > k+m|X > k) \to 1$. In accordance with Corollary C.1, the following inequality holds:

$$\log|\mathcal{V}|\Delta\text{Conf} \geq p_k \log \frac{p_k(|\mathcal{V}|-k)}{1-\sum_i^k p_i} + (1-\sum_i^{k-1} p_i)\log\frac{1}{p_k} + (1-\sum_i^{k-1} p_i)\log\frac{(1-\sum_i^k p_i)}{|\mathcal{V}|-k}.$$

Moreover, based on Corollary C.2, we can further derive:

$$\log|\mathcal{V}|\Delta\text{Conf} \geq p_k \log \frac{p_k(|\mathcal{V}|-k)}{1-\sum_i^k p_i} + p_k \log\frac{(1-\sum_i^k p_i)}{|\mathcal{V}|-k} + p_k\log\frac{1}{p_k} \geq p_k \log 1 = 0.$$

□

Detailed mathematical proof in Corollary 3.4 is as follows.

Consider

$$H(X) = -\sum_{i=1}^{k} P_\theta(x_i)\log P_\theta(x_i) - \sum_{k+1}^{|\mathcal{V}|} P_\theta(x_i)\log P_\theta(x_i).$$

As $P_\theta(X > k+m|X > k) \to 1$,

$$-\sum_{k+1}^{|\mathcal{V}|} P_\theta(x_i)\log P_\theta(x_i) \to 0. \tag{7}$$

We define the binary random variable

$$B(u) = \begin{cases} 1 \text{ if } X \leq u \\ 0 \text{ if } X > u. \end{cases}$$

.

Then, we can express $H(X)$ in terms of $B(u)$ as:

$$\begin{aligned} H(X) &= H(X \mid B(u)) + I(X; B(u)) \\ &= H(X \mid B(u)=1)\Pr\{B(u)=1\} + H(X \mid B(u)=0) \times \Pr\{B(u)=0\} + I(X; B(u)) \\ &= H(X(u))\Pr\{B(u)=1\} + H(X \mid B(u)=0) \times \Pr\{B(u)=0\} + I(X; B(u)). \end{aligned}$$

As $u \to k$, we have $\Pr(B(u)=1) \to 1$, leading to $H(B(u)) \to 0$. Consequently, $I(X; B(u)) \to 0$ due to $I(X; B(u)) \leq H(B(u))$.

Additionally, $H(X|B(u)=0)\Pr(B(u)=0)$ can be simplified as:

$$
\begin{aligned}
&H(X \mid B(u)=0)\Pr\{B(u)=0\} \\
&\quad = -\sum_{i=k+1}^{\infty} P_\theta(x_i)\log\frac{P_\theta(x_i)}{\Pr\{B(u)=0\}} \\
&= -\sum_{i=k+1}^{\infty} P_\theta(x_i)\log P_\theta(x_i) - \log\Pr\{B(u)=0\}) \\
&= -\sum_{i=k+1}^{\infty} P_\theta(x_i)\log P_\theta(x_i) + \left(\sum_{i=k+1}^{\infty} P_\theta(x_i)\right)\log\Pr\{B(u)=0\} \\
&= -\sum_{i=k+1}^{\infty} P_\theta(x_i)\log P_\theta(x_i) + \Pr\{B(u)=0\}\log\Pr\{B(u)=0\}.
\end{aligned}
$$

As $u \to k$, we have $\Pr(B(u)=0) \to 0$ and Equation (7), then both terms lead to zero. In conclusion, we obtain $H(X) = H(X(k)) + \epsilon$, where ϵ is a smaller number.

D. Hyperparameter

Choosing hyperparameters seems a challenging task, and the process of our parameter selection is illustrated in Table 10 and Table 11. Here are some interesting things we found:

- The top-k and top-p algorithms enhance diversity by expanding the truncation space, leading to higher MAUVE values. This occurrence can be attributed to the ability of LMs to sample certain low-probability tokens, which are proximate to the probabilities preferred by humans. This proximity results in a closer match between the two curves illustrated in Figure 1(a). However, it is crucial to recognize that the concurrent increase in diversity corresponds to a decrease in coherence.
- Mirostat excels in effectively managing text coherence by controlling model perplexity. The ease of controlling parameters further enhances its utility, even though some of the observed increase in coherence may be attributed to the rise in indices resulting from repetition. Additionally, our investigation reveals that Mirostat, when applied to the Llama2-7B model, maintains high coherence and ensures a certain level of diversity.

E. Instrcution

In Section 5.5, we use the following instruction, where the sentence is derived from the prompt in WritingPompts.

```
<s>[INST] <<SYS>>
You're a writer. You will write a story given my prefix.
<</SYS>>

Prefix: {sentence} [/INST]
```

In Section 5.6, we use the following instruction, where the problem is derived from the dataset GSM8K and MultiArith.

```
<s>[INST] <<SYS>>You are a help assistant and a math expert.
Please solve the following question and directly return me the answer.<</SYS>>
    Problem: {sentence}
    Let's think step by step\n[/INST]
```

F. Human Evaluation

The human evaluation adheres to the principles of blind review, ensuring that the evaluator remains unaware of the decoding algorithms employed in the two generations or the human writing in advance. The evaluator determines the superior outcome based on three guiding principles. Our application for human evaluation is illustrated in Appendix F.

Table 10. Optimal hyperparameter selection for various decoding algorithms in the GPT2-XL model. The line where the bold is located designates the optimal parameter value. We repeat the sampling process five times for each prefix and report the average score.

Decoding	WritingPrompts						WikiText-103					
	rep-2	**rep-3**	**rep-4**	**Diversity**	**MAUVE**	**Coherence**	**rep-2**	**rep-3**	**rep-4**	**Diversity**	**MAUVE**	**Coherence**
Human	7.70	1.82	0.79	90.06	–	29.57	5.14	1.02	0.33	93.66	–	63.49
$k = 5$	32.86	24.03	19.55	49.37	23.94	53.16	19.18	10.37	6.90	70.40	82.29	62.57
$k = 10$	19.60	11.83	8.71	69.82	33.65	50.04	11.50	4.44	2.44	83.41	82.49	62.25
$k = 15$	14.91	8.35	6.16	76.86	44.92	49.51	9.90	3.64	2.01	86.24	**88.75**	61.41
$k = 20$	12.23	6.26	4.35	81.22	**48.25**	47.87	8.52	2.78	1.48	88.51	88.67	61.61
$p = 0.80$	25.39	18.79	15.90	61.61	30.29	49.13	12.98	5.94	3.84	80.89	77.74	61.11
$p = 0.85$	19.78	13.14	10.80	70.02	35.10	48.20	11.23	4.87	3.02	83.73	79.61	61.46
$p = 0.90$	15.16	9.14	7.07	76.96	38.52	47.15	9.57	3.67	2.17	86.62	84.46	61.13
$p = 0.95$	12.21	7.04	5.53	81.43	**51.69**	47.35	7.70	2.70	1.49	89.24	**88.50**	60.87
mirostat-2.0	59.05	52.33	47.75	17.26	13.75	50.21	45.09	35.99	30.57	31.43	**56.59**	60.34
mirostat-3.0	58.73	51.91	47.27	17.82	16.52	49.63	45.13	36.09	30.61	31.27	55.29	60.73
mirostat-4.0	58.62	51.84	47.17	17.69	16.30	49.28	45.14	36.07	30.60	31.11	56.56	61.01
mirostat-5.0	58.55	51.70	47.04	18.17	**16.91**	49.71	45.46	36.46	31.03	31.11	53.55	61.09
$\tau = 0.90$	15.24	9.24	7.25	76.74	38.10	46.64	9.38	3.60	2.14	86.94	85.35	61.27
$\tau = 0.92$	14.02	8.09	6.27	78.65	**43.92**	46.63	9.08	3.35	2.06	87.47	84.99	61.00
$\tau = 0.95$	12.05	6.96	5.32	81.47	38.01	46.99	8.02	2.73	1.52	89.10	85.39	60.89
$\tau = 0.99$	9.22	4.63	3.64	85.42	40.57	46.59	6.74	1.93	0.95	91.13	**90.07**	59.93
$\eta = 0.004$	12.33	7.08	5.66	80.95	41.86	46.69	7.83	2.63	1.48	89.22	**90.40**	61.10
$\eta = 0.002$	9.61	4.98	3.63	85.42	47.24	45.60	7.58	2.75	1.80	89.81	82.18	59.55
$\eta = 0.0009$	9.10	4.58	3.41	86.58	**59.34**	45.67	6.45	1.75	0.98	91.38	87.17	60.33
$\eta = 0.0006$	8.99	4.38	3.67	85.98	46.23	44.38	6.44	1.72	0.82	91.64	86.45	60.37
$\epsilon = 0.0005$	5.46	3.16	2.75	91.38	47.19	36.38	4.82	1.55	1.00	93.81	83.72	56.58
$\epsilon = 0.001$	7.07	3.85	3.11	89.32	**55.19**	37.72	5.22	1.51	0.93	93.00	**84.54**	56.39
$\epsilon = 0.005$	40.69	33.57	29.95	39.80	11.02	47.51	21.10	12.50	9.10	67.63	69.27	61.30
$\epsilon = 0.01$	71.14	67.75	65.28	9.81	2.89	50.71	45.17	37.82	33.96	34.02	23.08	61.04

- Consistency: The extent to which the produced text maintains semantic coherence with the given context.
- Smoothness: The degree to which the generated text flows seamlessly and is readily comprehensible.
- Richness: The presence of diverse and captivating content in the generated text

G. Case Study

Given the prefix sampled from WikiText-103, which provides relevant enough context, we give the truncated human continuation and the generation of our proposed adaptive decoding. Cases can be found in Table 12 and Table 13.

Table 11. Optimal hyperparameter selection for various decoding algorithms in the Llama2-7B model. The line where the bold is located designates the optimal parameter value. We repeat the sampling process five times for each prefix and report the average score.

Decoding	WritingPrompts						WikiText-103					
	rep-2	rep-3	rep-4	Diversity	MAUVE	Coherence	rep-2	rep-3	rep-4	Diversity	MAUVE	Coherence
Human	7.70	1.82	0.79	90.06	–	29.57	5.14	1.02	0.33	93.66	–	63.49
$k = 5$	22.69	14.76	11.10	66.96	39.66	27.15	12.14	5.56	3.20	81.69	63.82	59.43
$k = 10$	14.21	7.26	5.05	79.94	45.34	27.44	9.35	3.70	2.07	86.51	74.15	59.51
$k = 15$	11.00	5.42	3.65	84.46	40.40	25.98	7.88	2.76	1.40	89.11	82.88	59.38
$k = 20$	9.62	4.19	2.94	86.29	**47.36**	26.25	7.44	2.55	1.40	89.54	**87.07**	59.27
$p = 0.80$	17.85	10.92	8.23	74.17	46.40	27.23	11.22	5.05	2.94	83.40	75.67	59.33
$p = 0.85$	13.51	7.66	5.58	80.26	44.91	27.55	9.64	4.01	2.29	85.84	79.15	58.94
$p = 0.90$	11.02	5.41	3.91	83.58	55.53	31.11	8.20	3.02	1.58	88.37	80.49	59.54
$p = 0.95$	8.39	4.32	3.26	87.96	**58.75**	26.99	7.13	2.52	1.32	90.12	**82.22**	58.89
mirostat-2.0	13.58	7.99	6.04	79.94	44.55	55.91	12.25	6.10	3.94	80.99	81.55	66.51
mirostat-3.0	13.46	7.85	5.88	80.04	**50.51**	54.83	12.36	6.14	3.99	80.72	**82.06**	65.87
mirostat-4.0	14.95	8.59	6.85	79.11	42.97	55.10	12.70	6.50	4.32	80.06	81.09	66.31
mirostat-5.0	15.08	9.82	7.88	77.82	45.22	54.84	12.49	6.23	4.06	80.53	78.36	66.46
$\tau = 0.90$	10.83	5.36	3.84	84.65	38.07	25.23	8.24	3.24	1.77	88.10	77.49	59.23
$\tau = 0.92$	9.37	4.37	3.38	86.86	33.88	27.80	7.68	2.80	1.46	89.17	82.78	58.79
$\tau = 0.95$	9.54	5.03	3.78	87.09	40.18	26.80	6.81	2.20	1.06	90.61	**87.75**	58.83
$\tau = 0.99$	8.67	4.53	4.13	88.93	**53.26**	25.76	6.00	1.98	1.00	91.75	80.68	58.50
$\eta = 0.004$	7.86	3.42	2.50	88.69	48.93	26.30	7.15	2.47	1.30	90.04	77.47	58.76
$\eta = 0.002$	7.17	3.58	2.62	89.76	48.34	26.88	6.41	2.21	1.17	91.15	81.49	58.76
$\eta = 0.0009$	7.10	3.04	2.14	90.42	61.30	26.78	5.68	1.72	0.85	92.41	86.14	58.76
$\eta = 0.0006$	6.69	2.93	2.42	90.43	**63.80**	26.35	5.87	1.73	0.76	92.04	**88.28**	58.75
$\epsilon = 0.0005$	7.21	2.84	1.74	89.54	**69.00**	35.40	5.90	1.78	0.79	92.00	87.18	58.78
$\epsilon = 0.001$	8.38	3.54	2.14	87.76	59.34	35.92	6.74	2.16	1.03	90.65	**87.81**	59.26
$\epsilon = 0.005$	19.22	11.44	8.43	70.75	58.23	42.35	10.42	4.61	2.69	84.76	76.91	60.03
$\epsilon = 0.01$	35.60	27.20	22.96	47.12	43.69	44.02	14.11	7.34	4.65	78.55	68.76	58.69

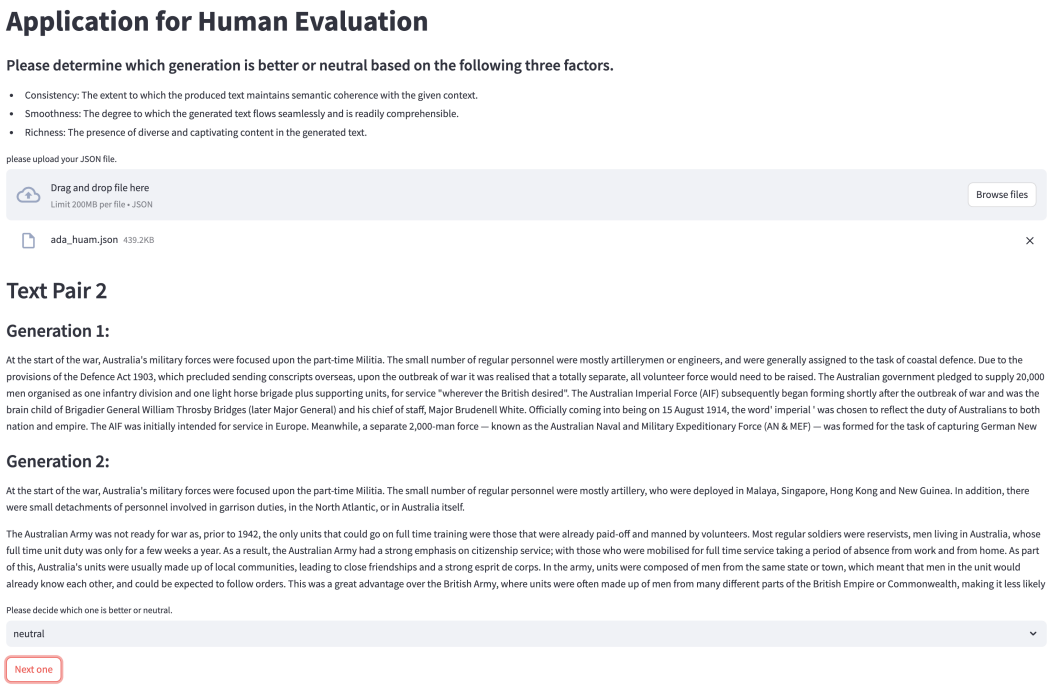


Figure 6. Our application for human evaluation.

Table 12. Case study: Text continuation with the GPT2-XL model. Entities are in blue fonts, logical relationships are in red, and coherent contents are highlighted in green.

Prefix: Following a period of rest in England, Little turned down a desk assignment and volunteered to return to action on the Western Front, joining Lieutenant Colonel Collishaw's
Human Continuation: No. 3 Squadron RNAS in March 1918. The unit evolved into No. 203 Squadron of the new Royal Air Force on 1 April, formed after the merger of the RNAS and the Royal Flying Corps. Now ranked captain, and again flying Sopwith Camels, Little gained a further nine successes, beginning with a Fokker Triplane on 1 April, and concluding with two kills in one day on 22 May, an Albatros and a DFW. During this stretch of victories, on 21 April 1918, he was brought down unharmed by Friedrich Ehmann. On 27 May, Little received reports of German Gotha bombers in the vicinity, and took off on a moonlit evening to intercept the raiders. As he closed with one of the bombers, his plane was caught in a searchlight beam and he was struck by a bullet that passed through both his thighs. He crash-landed in a field near Nœux, and bled to death before he was discovered the following morning by a passing gendarme. Little's skull and ankle had also been fractured in the impact; his body was identified by his friend and fellow ace, Charles Dawson Booker.
Adaptive decoding: 49th Rifles. On his first day back, he led a cavalry charge which was repulsed in the village of Culmore Wood in June 1916, while advancing across an open plain on the company's way to Vimy Ridge. In late 1916, He had been promoted from sergeant and given a smaller objective position in the near-by village of Meadowhall. His company was soon attacked by a cavalry squad of such ferocity that the archers and machine gunners could have opened up at once. Little was bitten, painfully, by a cavalry lance. He received the nicknames "Noire" and "The Snake", because of his long knife, thin face, and reptilian nature. He later hired a pet donkey to pull his belongings back to camp at the end of each day's march. In late May 1917, three weeks after the Heed affair, Little was deployed for the first time in front of the Hindenburg Line when, as part of the Anglo-French alliance, his squad was to capture and destroy German flak guns in Belgium. As if on cue, two machine gun crews came through a broken field and attacked Little's men, who retreated through a canal.
Prefix: The total population size of the African river martin is unknown. In the late 1980s, it appeared to be common, if local, and large numbers were
Human Continuation: seen on migration in Gabon. However, it is particularly poorly studied in the Democratic Republic of the Congo (DRC), and it is not known if there is any relationship between the birds breeding in the DRC and those breeding in coastal areas of Gabon and Congo. Several hundred birds were seen at Conkouati-Douli National Park in Congo in 1996 and a flock of 15,000 birds was seen in Gabon in 1997. Investigations in Gabon in September 2003 extended the known range of this martin. More than 300 birds were found with hundreds of rosy bee-eaters in the Omboué area and the newly established Iguéla National Park, and a mixed flock with rosy bee-eaters at Loango National Park in Gabon was estimated at 100,000 birds in total. In January 2010, 250 martins were seen at a new location in Bakoumba, and single birds were observed in hirundine flocks near Moanda, at Mounana, and at Lekoni.
Adaptive decoding: seen during spring breeding in the Cape province of South Africa. The species is also common in South America, but these differences between species and range regions is unclear. The African river martin is a robust and broad-bodied species that has a dark body and light face, which enhances its camouflage. Its chest is often painted or spotted with black. The coloration and markings can differ between individuals. They have very large ears and a long tail, and in female martins have a pattern that resembles a dorsal stripe, which appears more yellow than brown when united with dark bars, thus giving them the common name of yellow-backed or yellow-fronted. The tail is large, and in males has a faint red pink spot at the base that is very difficult to see unless the animal is under a light. The neck varies from strong to weak, and males often have longer teeth.
Prefix: The water temperature of Briar Creek near its source ranges from 35.96 to 64.94°F (2.20 to 18.30°C),
Human Continuation: with a median of 51.44°F (10.80°C). This is the highest minimum temperature of any studied location in the watershed. Near West Branch Briar Creek's confluence with Briar Creek, the latter creek has a temperature range of 32 to 67.46°F (0.00 to 19.70°C) and a median of 49.28°F (9.60°C). This is the lowest minimum temperature of any studied location in the drainage basin. Near its mouth, the temperature ranges from 32.18 to 80.96°F (0.10 to 27.20°C). The median temperature is 57.56°F (14.20°C). On West Branch Briar Creek, the temperature is a minimum of 35.6°F (2.0°C), a median of 57.38°F (14.10°C), and a maximum of 71.6°F (22.0°C). Two miles downstream of East Branch Briar Creek's headwaters, the median temperature is 49.73°F (9.85°C) and the range is 32.18 to 71.42°F (0.10 to 21.90°C).
Adaptive decoding: with a seasonal mean range of 40.52°F (3.68°C). Seasonal variation in temperature, as well as production of sulfides, can significantly affect food source quality, quality of water, and the quality and quantity of sediment that may be trapped in the porous pipes and culverts that lead into groundwater. Relatively constant temperatures between 40°F (4°C) and 55°F (13°C) are common in the watershed. Temperatures above these extremes may occur occasionally when the watershed is dry, but the average annual maximum temperature in Briar Creek is 52.92°F (12.55°C). Average summer temperatures on the creek's western slopes are generally less than 60°F (15°C) throughout most of the year. Sustaining temperature above 60°F (15°C) is usually limited to the summer months and are relatively infrequent. Briar Creek drains nine watersheds with areas in the northern half of the watersheds containing more than 70% of the plants (wheat, barley, rice, rye and bulgur), but the predominant tree species (huckleberry, aspen, birch, and pine) is oak.

Table 13. Case study: Text continuation with the Llama2-7B model. Entities are in blue fonts, logical relationships are in red, and coherent contents are highlighted in green.

Prefix: Liszt found his method through two compositional practices, which he used in his symphonic poems. The first practice was cyclic form, a
Human Continuation: procedure established by Beethoven in which certain movements are not only linked but actually reflect one another's content. Liszt took Beethoven's practice one step further, combining separate movements into a single-movement cyclic structure. Many of Liszt's mature works follow this pattern, of which Les préludes is one of the best-known examples. The second practice was thematic transformation, a type of variation in which one theme is changed, not into a related or subsidiary theme but into something new, separate and independent.
Adaptive decoding: form that repeats the thematic material several times in different musical settings, as with the work of Haydn and Beethoven. It also was used in works such as Wagner's opera cycle, Der Ring des Nibelungen and Brahms' Alto Rhapsody. Liszt's second practice was program music, which uses musical form in order to support programmatic concepts. Liszt applied both compositional practices to his symphonic poems, using cyclic form to construct musical episodes that reflect programmatic ideas. The music reflects the program throughout the piece.
Prefix: Rebellions can occur if the loyalty of a particular province falls too low, with a rebel army appearing in the province to attempt to assume control
Human Continuation: from the owners. Civil wars may also take place if several generals commanding large armies have sufficiently low loyalty. In the event of a civil war, the player is given the choice to back either the current rulers or the rebels. It had been planned to allow other factions who had established a prior claim to the throne by marriage to princesses to join in a civil war to claim the throne for themselves; however, this was never implemented. Naval warfare is carried out upon the campaign map, where ships can be built and organised into fleets. These fleets can be used to control the game's sea regions and form sea lanes, allowing trade and troop movement between provinces that have constructed a port. Fleets can engage in sea battles with foreign fleets, although unlike land battles these are resolved by the computer. Religion plays an important aspect in Medieval: Total War, with religious differences between the Catholic, Orthodox and Muslim factions affecting diplomacy and population loyalty.
Adaptive decoding: . If the loyalty falls too far, the province will secede, and become a separate, unallied nation (this may sometimes be undesired, so it is important to know how to increase loyalty and to always pay attention to the needs of all provinces). As a province is loyal, its loyalty will be decreased by any war that the province takes part in, and will be increased by having any kind of benefit in the province. As a result of the fall of the West Roman Empire, most of the Western Mediterranean provinces were either conquered or claimed by the Christian Roman States, or by Germanic tribes such as the Visigoths, Ostrogoths or Vandals. The Eastern Mediterranean however is largely in the hands of Arabic Caliphates, or the Empire of Japan. The Western Mediterranean is the primary area where player wars begin, although it is possible to war in the eastern mediterranean, the arctic, or most recently, in the Indian Ocean. The Western Mediterranean however is also the richest area in the game, with a large amount of resources and a large number of wealthy cities which can be attacked.
Prefix: In 1947 Bessie was elected to the Labour Party's National Executive Committee (NEC). She was generally identified with the left
Human Continuation: wing of the party, and was for a time associated with a grouping known as the "Socialist Fellowship", which espoused a programme of colonial freedom, workers' control and reduced arms expenditure. She resigned from the Fellowship in 1950, along with fellow-MPs Fenner Brockway and Ellis Smith, when it condemned the United Nations intervention in the Korean War. She continued her wholehearted campaigning on behalf of the poorest in the country, pleading with parliament to "remember the queues outside the Poor Relief offices", and castigating the "New Look" fashion of 1948 as wasteful, "the ridiculous whim of idle people". Bessie's fiery reputation did not harm her electorally; in the February 1950 general election, with the Exchange constituency greatly increased by boundary changes, her majority rose to 5,344. Nationally Labour lost 76 seats, and its parliamentary majority was reduced to five. Attlee's second government was short-lived;
Adaptive decoding: wing of the Labour Party. She was also active in organisations supporting Jewish refugees from the Nazis. With the help of another Jewish member of the Labour Party, Maurice Jacobson, Bessie had her own radio programme (Bessie Braddock's Book) and also became a popular speaker at meetings. In 1947 she was approached by the National Union of General and Municipal Workers (NUGM) and asked if she would run for parliament in an election. In the general election of 1950 Bessie Braddock was elected as Member of Parliament for Liverpool Exchange. She stood for the National Union of General and Municipal Workers, Labour Co-Operative and the Socialist Medical Association (of which she was a founder member). Her election campaign became more and more left wing. In Parliament she was very outspoken and unpredictable, using colourful language in her speeches. A political opponent once described her as "the only man in the House of Commons". In 1954 she was appointed as parliamentary private secretary to Hugh Gaitskell.