A Thorough Examination of Decoding Methods in the Era of LLMs

Anonymous ACL submission

Abstract

Decoding methods play an indispensable role in converting language models from next-token predictors into practical task solvers. Prior research on decoding methods, primarily focusing on task-specific models, may not extend to the current era of general-purpose large language models (LLMs). Moreover, the recent influx of decoding strategies has further complicated this landscape. This paper provides a comprehensive and multifaceted analysis of various decoding methods within the context of LLMs, evaluating their performance, robustness to hyperparameter changes, and decoding speeds across a wide range of tasks, models, and deployment environments. Our findings reveal that decoding method performance is notably task-dependent and influenced by factors such as alignment, model size, and quantization. Intriguingly, sensitivity analysis exposes that certain methods achieve superior performance at the cost of extensive hyperparameter tuning, highlighting the trade-off between attaining optimal results and the practicality of implementation in varying contexts.

1 Introduction

The advent of large language models (LLMs) (OpenAI, 2022, 2023; Touvron et al., 2023a,b, *inter alia*) has ushered in a new era of natural language processing (NLP). These models are trained to predict the next token on massive corpora, empowering them with extraordinary multitasking capabilities. This enables them to perform almost all NLP tasks through the lens of text generation, distinguishing them from traditional task-specific models.

Decoding methods, which are the bridge between *next-token predictors* and *text generators*, play an integral role in transforming LLMs into practical task solvers. Recent studies have shown that the choice of decoding methods can substantially impact the performance of LLMs (O'Brien and Lewis, 2023; Chuang et al., 2023). However, these studies often focus on a narrow aspect (e.g., factuality (Chuang et al., 2023)) and a limited set of similar tasks (e.g., math problem solving (Li et al., 2023b)). Notably, Ippolito et al. (2019); Wiher et al. (2022) provide a comparative analysis of various decoding methods using task-specific language models. They find that deterministic decoding methods (e.g., beam search) perform better than stochastic decoding methods (e.g., top-p sampling (Holtzman et al., 2020)) in closed-ended generation tasks such as machine translation, while the inverse is true for open-ended generation tasks such as story generation. However, their findings are confined to traditional task-specific models prior to the advent of LLMs. It is uncertain whether their conclusions still hold for general-purpose LLMs. In addition, a plethora of new decoding methods (Su et al., 2022; Li et al., 2023b; Yang et al., 2023; Meister et al., 2023; Hewitt et al., 2022; Basu et al., 2021) have been proposed afterward, each claiming to outperform the previous state-of-the-art in particular tasks. Nevertheless, today's most performant LLMs such as ChatGPT and GPT4 (OpenAI, 2022, 2023) only provide APIs for temperature and top-p sampling, seemingly overlooking the potential benefits of other advanced decoding methods.

The above observations raise a natural question: what is the best practice for choosing decoding methods in the era of LLMs? A thorough analysis of decoding methods is essential for researchers and practitioners to understand the strengths and weaknesses of different decoding methods and to choose the one that best fits their needs. Our work fills this gap by providing a comprehensive study of the *performance*, *robustness*, and *speed* of various decoding methods across a wide range of different tasks, models, and deployment environments. We also provide in-depth analyses to uncover the underlying reasons for the observed results. Our key findings include the following:

- **Overall** The optimal decoding method depends on the task, the model, and the priority (e.g., performance vs. robustness vs. speed) in hand. There is no short guideline. The complexity of our results calls for more comprehensive evaluations in future research on decoding methods and careful consideration for practitioners.

- **Performance** The best-performing methods depend on the task at hand. However, some general rules about the divide between different decoding methods still persist in the era of LLMs. Generally, closed-ended tasks favor deterministic methods, while open-ended tasks prefer stochastic methods (§4.1), especially with unaligned models. The performance gap between different decoding methods can be narrowed with alignment. We also provide explanations to understand these phenomena. Moreover, it is also observed that stochastic methods with self-consistency can surpass deterministic ones, albeit requiring multiple runs (§5.1).
- **Robustness** The optimal hyperparameters for each decoding method vary according to the model, task, and quantization setting. Some methods achieve superior performance at the cost of exhaustive dataset-specific hyperparameter searches but fail to maintain the superiority when the hyperparameter is fixed. This highlights the *performance-sensitivity* trade-off because LLMs are often confronted with diverse user prompts (§4.2).
- **Speed** Stochastic decoding and the recently proposed deterministic method, frustratingly simple decoding (FSD) (Yang et al., 2023), can achieve a similar decoding speed to greedy search. In contrast, beam search, diverse beam search and other advanced deterministic methods show markedly slower speeds relative to greedy search, with the discrepancy in speed becoming more conspicuous as the length of generation increases for some of those methods (§4.3).

2 Decoding Methods

Modern LLMs typically generate text in a left-to-right, token-by-token fashion. For each prefix, the model computes a probability distribution of the next token over a fixed vocabulary. A decoding method defines how the generated token sequence is derived from these probability estimations. We consider decoding methods ranging from deterministic to stochastic. Each method is briefly reviewed below, with detailed descriptions in Appendix A. The hyperparameter search range of each method is guided by recommendations from relevant literature and common practices.

2.1 Deterministic Methods

Greedy Search selects the token with the highest probability at each time step.

Beam Search (BS) (Freitag and Al-Onaizan, 2017) maintains a beam of the k most probable sequences at each time step, where the hyperparameter k is referred to as the beam width. We consider beam sizes 4 and 8 in our experiments.

Diverse Beam Search (DBS) (Vijayakumar et al., 2018) is a variant of beam search that divides the k most probable sequences into G groups and incorporates a diversity term to maximize inter-group diversity. In our experiments, we configure various (k, G) pairs of (4,2), (4,4), (8,2), (8,4).

Contrastive Search (CS) (Su et al., 2022) uses a look-ahead mechanism and penalizes tokens compromising the isotropy of the LM's latent space. We search the penalty degree from $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$ in our experiments.

Contrastive Decoding (CD) (Li et al., 2023b) searches for tokens that maximize the probability difference between the LLM and a weaker amateur model. We search the the strength of the amateur penalty from $[0.1, 0.3, 0.5, 0.7, 0.9]$.

Frustratingly Simple Decoding (FSD) (Yang et al., 2023) exploits the contrasts between the LLM and an auxiliary anti-LM constructed based on the current prefix. There are two variants of FSD: FSD and FSD-d depending on whether the anti-LM is implemented as a vectorized or discrete n-gram model. We search penalty degree from $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$.

DoLa (Chuang et al., 2023) obtains the next-token distribution by contrasting the logits differences between the last layer and a premature layer. The premature layer is dynamically selected from a pre-specified set of layers. Following Chuang et al. (2023), we test two sets of layers: even-numbered layers from $[0, 16)$ and from $[16, 32)$ respectively.

2.2 Stochastic Methods

Temperature Sampling samples tokens from the estimated next-token distributions. The skewness of distributions can be controlled using a temperature hyperparameter τ. We conduct our experiments for τ within the range of 0.1 to 0.9, incrementing in value of 0.1.

Top-p Sampling (Holtzman et al., 2020) only considers the minimal set of most probable tokens that cover a specified percentage p of the distribution. We examine across various p thresholds, specifically $[0.8, 0.85, 0.9, 0.95, 1]$.

Top-k Sampling (Fan et al., 2018) only samples from the top-k probable tokens. We explore a range of k values, specifically $[5, 10, 20, 50, 100]$.

η-Sampling (Hewitt et al., 2022) truncates words whose probabilities are below an entropy-dependent threshold. The hyperparameter η is searched from [3e-4,6e-4,9e-4,2e-3,4e-3].

Mirostat Sampling (Basu et al., 2021) directly controls the perplexity rate of the generated text during sampling from top-k tokens (k is determined automatically). We test across a range of log of perplexity values τ within $[2.5, 3, 4, 5]$.

Typical Sampling (Meister et al., 2023) sorts the vocabulary according to the differences between the distribution entropy and the token probabilities. In our experiments, we vary the coverage threshold p across the values $[0.2, 0.9, 0.92, 0.95]$.

3 Evaluation Setup

3.1 Datasets

Our evaluation spans a variety of tasks.

Coding is an important application of LLMs, facilitating the integration with external tools. We use HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021), reporting pass@1 accuracy.

Math Problem Solving is critical for LLMs, enabling them to aid users in numerical reasoning tasks. We employ GSM8K (Cobbe et al., 2021) for this purpose and report accuracy.

Summarization assists users in capturing the essence of a text. We use CNN/DailyMail (CNN/DM) (Hermann et al., 2015) and XSUM (Narayan et al., 2018), measuring performance with RougeL (Lin, 2004).

Translation is a crucial NLP task to overcome linguistic barriers, thereby facilitating global communication. We benchmark it using four directions of WMT22 (Bojar et al., 2017) and assess the translation quality via BLEU (Papineni et al., 2002).

Commonsense Reasoning is a key perspective of LLMs for addressing real-world problems. We assess this using CommonsenseQA (CQA) (Talmor et al., 2019) and StrategyQA (SQA) (Geva et al., 2021), reporting accuracy.

Factual Knowledge is crucial for fulfilling users' informational needs. We measure this using FActScore (Min et al., 2023), reporting on the proportion of correctly generated atomic facts.

Instruction Following reflects the proficiency in responding to diverse user instructions. We use AlpaceEval (Li et al., 2023c) to compare model performances, using pairwise Win Rate against the reference model, Text-Davinci-003.

Open-ended Text Generation measures the model's capability to produce fluent and coherent content. We utilize datasets including Book (Zhu et al., 2015), Wikinews[1], and Wikitext (Merity et al., 2017), and evaluate using MAUVE (Pillutla et al., 2021). Notably, open-ended text generation is the primary focus for many recent decoding methods.

For detailed task descriptions and prompts, see Appendix B and Appendix C. Generally, higher scores in respective metrics indicate better performance.

3.2 Models

We primarily experiment with the Llama-2 family, comprising Llama2 and Llama2-chat (Touvron et al., 2023b), representing unaligned and aligned models, respectively. Additional tests include other popular LLMs: MPT (Team, 2023), CodeLlama (Rozière et al., 2023), Qwen (Bai et al., 2023), Mistral (Jiang et al., 2023), Deepseek-MoE (Dai et al., 2024) and Llama3 (AI@Meta, 2024), along with their aligned counterparts are detailed in Appendix D. Unaligned models are not tested on AlpaceEval and FActScore due to their limited instruction-following capabilities. Owing to poor performance for WMT22 with Llama2-Chat, its performance is measured only on the unaligned model. Unless otherwise specified, we employ half-precision (FP16) for model inference.

4 Experimental Results

We perform a thorough evaluation of various decoding methods, assessing them from three critical dimensions. Initially, our analysis centers on the efficacy of these methods across a diverse range of tasks and models. Then, we delve into hyperparameter sensitivity and decoding efficiency.

4.1 Performance Analysis

We present the performance of decoding methods on unaligned and aligned Llama2-7B models (Llama2-7B and Llama2-7B-Chat respectively) in Table 1. The reported results for each method

[1] http://www.wikinews.org

Model	Dataset	Metric	Deterministic Methods								Stochastic Methods					
			Greedy	BS	DBS	CS	FSD	FSD-d	CD	DoLa	Temp	Top-p	Top-k	η	Miro	Typical
Llama2-7B	HumanEval	Pass@1	12.80	15.24	15.24	14.63	15.24	15.24	14.02	15.24	15.24	9.15	8.54	9.15	7.93	9.76
	MBPP		17.80	19.40	18.40	17.40	19.20	21.20	18.20	18.40	17.20	14.80	10.20	9.40	7.80	12.00
	GSM8K	Acc	13.87	17.21	17.74	14.63	16.83	16.60	17.21	15.39	16.30	12.96	9.10	8.64	7.96	13.04
	XSUM	R-L	27.21	21.88	24.65	27.53	27.75	27.88	27.36	25.92	27.14	22.34	22.10	20.45	20.23	21.33
	CNN/DM		23.43	20.69	21.64	23.25	23.39	24.05	23.73	22.64	23.40	20.52	20.90	18.63	18.02	19.13
	De⇒En	B-4	28.80	30.14	28.71	28.63	28.52	28.82	28.40	25.45	28.55	22.72	20.30	18.44	18.00	20.00
	En⇒De		22.63	23.99	23.52	22.74	22.54	22.63	22.30	19.82	22.57	16.14	14.32	12.28	11.62	13.34
	Zh⇒En		19.44	20.11	18.90	19.56	19.71	20.05	19.68	17.06	19.26	13.35	12.02	10.26	9.60	10.78
	En⇒Zh		15.15	14.50	14.67	15.27	15.21	15.37	14.57	13.09	15.21	11.61	11.27	11.50	7.89	9.94
	CQA	Acc	62.90	64.37	64.21	63.72	64.05	63.72	62.65	62.00	63.72	56.51	49.47	47.17	46.11	52.91
	SQA		60.76	62.25	61.50	60.54	62.90	60.89	63.74	61.94	61.20	58.71	58.09	58.27	58.44	58.05
	Wikinews	MAUVE	40.10	41.33	32.02	96.66	96.42	98.40	85.17	94.44	95.40	95.19	96.47	97.48	98.51	97.67
	Wikitext		23.47	27.41	22.78	93.38	92.14	92.93	85.86	85.39	94.54	96.62	96.67	93.66	93.18	93.29
	Book		13.10	17.54	10.18	88.41	89.07	86.69	73.30	80.54	90.62	95.99	94.84	95.31	94.25	93.98
Llama2-7B-Chat	HumanEval	Pass@1	12.80	14.02	13.41	13.41	15.24	13.41	14.02	15.85	14.63	13.41	14.02	12.20	12.80	12.80
	MBPP		17.20	21.60	21.20	17.40	17.80	17.80	17.40	18.00	20.00	17.60	16.00	17.00	16.00	18.00
	GSM8K	Acc	24.79	28.81	26.91	25.70	25.40	24.56	26.46	22.14	25.47	24.26	24.41	25.25	23.20	24.11
	XSUM	R-L	16.42	16.96	16.78	16.70	16.63	16.52	16.49	8.84	16.51	16.44	16.28	16.44	15.77	16.77
	CNN/DM		22.59	23.71	23.54	22.54	22.40	22.64	22.65	16.92	22.71	22.67	22.03	22.34	20.60	22.42
	CQA	Acc	50.61	52.99	52.83	51.43	52.66	51.11	52.01	52.74	53.56	53.15	51.76	51.52	52.66	52.91
	SQA		59.89	60.41	60.59	59.97	60.32	60.37	60.19	59.62	60.19	60.28	60.80	60.10	59.41	59.14
	Wikinews	MAUVE	58.34	71.01	74.13	70.42	76.74	81.84	74.33	63.99	83.84	76.76	79.65	72.24	70.02	72.32
	Wikitext		77.69	87.20	90.27	80.16	95.10	90.47	84.59	38.76	80.45	80.51	85.63	87.52	83.48	89.77
	Book		80.65	94.89	93.78	90.81	94.75	92.00	95.96	57.70	96.55	91.50	93.48	89.95	92.95	93.87
	FActScore	Score	44.74	47.80	47.29	46.09	46.09	46.93	46.11	36.37	45.06	44.78	44.11	46.81	44.06	46.55
	AlpacaEval	WinRate	76.40	77.89	78.63	79.88	80.50	79.88	81.24	55.40	77.76	78.01	77.39	79.38	75.53	78.26

Table 1: Results on Llama2-7B and Llama2-7B-Chat. Cells are colored by performance, from low to medium to high performance. The corresponding hyperparameters for each decoding method are listed in Appendix E.

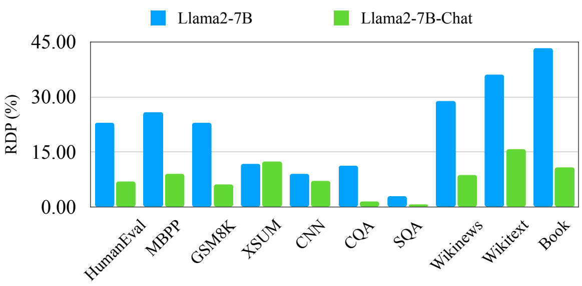

Figure 1: Relative deviation percentage (RDP) for each task on Llama2-7B and Llama2-7B-Chat.

are obtained by utilizing the best hyperparameters tuned for each specific dataset.

For unaligned models, deterministic methods generally perform better than stochastic methods on all tasks except open-ended text generation. As shown in the upper block of Table 1, for the unaligned Llama2-7B model, the top-performing decoding methods on closed-ended tasks (coding, math problem solving, summarization, translation, and commonsense reasoning) are frequently among deterministic methods. On the other hand, stochastic methods often struggle with the worst performance. Specifically, BS, FSD-d, and FSD rank in the top 3 (indicated in orange) in 8, 7, and 7 out of 11 datasets, respectively. Conversely, mirostat, η, and typical sampling are among the least effective three methods (highlighted in blue) in 10, 10, and 7 datasets, respectively. For open-ended text generation (Wikinews, Wikitext, and Book), greedy, BS, and DBS exhibit notably lower MAUVE scores than other methods. The above observations on the disparity of deterministic and stochastic methods are consistent with the findings for conventional task-specific models (Wiher et al., 2022): stochastic methods are favorable in open-ended tasks, while heavily disfavored in others.

→***Phenomenon Analysis.*** Through a careful case study, we find that the outputs of greedy, BS, and DBS contain a considerable amount of repetitive content on open-ended text generation tasks. This suggests that the advanced unaligned LLMs still suffer from the degeneration issue (Holtzman et al., 2020; Li et al., 2023a). Recent deterministic methods (CS, FSD, FSD-d, CD, and DoLa), which are designed to alleviate the degeneration issue, achieve much better results, performing only slightly worse than stochastic methods. For closed-ended tasks, deterministic approaches are better suited to producing consistent and accurate results as diversity is not a primary concern.

Aligned models are less dependent on decoding methods than unaligned models. For the unaligned Llama2-7B model, there is a clear separation between the highest- and lowest-performing methods. For instance, on MBPP, the highest performance is at 21.20% by FSD-d, in stark contrast to the lowest at 7.80% by mirostat sampling. However, this distinction becomes less pronounced for the aligned Llama2-7B-Chat model. Specifically, on MBPP, the top performance peaks at 21.60% while the lowest is at 16.00%, showcasing a narrowed performance range.

To further substantiate this, we compute the average μ and standard deviation σ of each dataset across different decoding methods. We report the *relative deviation percentage* (RDP) $\frac{\sigma}{\mu} \times 100\%$, of which a lower value signifies less performance variation across different decoding method choices. The results are depicted in Figure 1. Generally, the aligned model (Llama2-7B-Chat) displays less pronounced variations compared to its unaligned counterpart (Llama2-7B), except in two summarization datasets (XSUM and CNN/DM) where the relative deviation percentages are close. This suggests that the choice of decoding method becomes less critical after the model is aligned. Additionally, we also notice that DoLa performs quite worse than other methods under Llama2-7B-Chat. We check its outputs and observe that DoLa fails to terminate its generation appropriately (see Appendix F).

→***Phenomenon Analysis.*** The potential reasons are as follows: i) The improved model confidence. As shown in Table 2, we report the average next-token prediction entropy of Llama2-Chat-7B and Llama2-7B on GSM8K, MBPP, and Wikinews. It can be seen that the entropy of the aligned model is substantially lower than that of the unaligned one. As the model becomes more confident (concentrating the probabilistic mass on a shortlist of tokens), there is less operating space for decoding methods. ii) The alleviated degeneration issue. We find that the aligned model produces much fewer repetitions even when using deterministic decoding methods such as greedy search. This inherent improvement, possibly due to the high-quality data with reduced repetition employed during the instruction tuning phase (Li et al., 2023a), makes those decoding methods that aim to mitigate the degeneration issue less useful. iii) The more structured writing style. The aligned model typically produces more well-organized responses (e.g., a list of points with explicit discourse markers). This structural coherence enhances the stability of the model's output and reduces the variations of stochastic decoding methods (Lin et al., 2023a).

Deterministic methods tend to generate fewer hallucinations and have better instruction-following abilities. The lower block of Table 1 also presents the results of the aligned model (Llama2-7B-Chat) on FActScore and AlpacaEval. For FActScore, the top three best-performing methods are all deterministic. For instance, beam search attains 47.80%, while mirostat and top-k sampling only achieve scores of 44.06% and 44.11%, respectively. These results indicate that the choice of decoding method has a considerable impact on the factuality of the generated text. The randomness in the selection process of stochastic methods may contribute to increased hallucinations. For AlpacaEval, the general instruction-following task, deterministic methods such as CS, FSD, and CD can outperform all stochastic methods. This observation challenges the prevailing common practice of employing stochastic methods, particularly temperature and top-p sampling, in LLMs. This suggests that deterministic methods are more reliable for tasks requiring high factual accuracy and precise adherence to instructions, warranting further exploration in future research.

Model	GSM8K	MBPP	Wikinews
Llama2-7B	1.05	1.21	2.37
Llama2-7B-Chat	0.27	0.39	0.52

Table 2: The entropy of Llama2-7B and Llama2-Chat-7B's generation results (top-p sampling with $p = 1.0$) on GSM8K, MBPP and Wikinews.

Among stochastic methods, temperature sampling generally performs better, particularly when using unaligned models. As evidenced in Table 1, temperature sampling generally outperforms other stochastic methods except for open-ended text generation. Specifically, on Llama2-7B, temperature sampling emerges as the top-performing stochastic method across all 11 closed-ended tasks. Similarly, under Llama2-7B-Chat, it takes the top position in 5 out of 9 closed-ended tasks. We find that the best results often come from a low temperature (e.g., $\tau = 0.1, 0.2$, see Table 27 in Appendix E), which renders temperature sampling more akin to deterministic decoding. It is worth noting that many previous studies (Fan et al., 2018; Holtzman et al., 2020; Meister et al., 2023; Hewitt et al., 2022) predominantly demonstrate the superiority of their proposed methods in the realm of open-ended text generation. However, our analysis reveals that temperature sampling markedly surpasses these methods in closed-ended generation tasks, thereby underscoring the necessity for more holistic evaluations across diverse tasks.

4.2 Hyperparameter Sensitivity

The results in Table 1 are obtained by searching for the optimal hyperparameter of each decoding method for each dataset. Nevertheless, hyperparameter search is time-consuming and may not be plausible for open-world applications where the target task is not known a priori. Therefore, we further explore a more realistic scenario in which each method uses a fixed hyperparameter across different datasets. To ensure a fair comparison that accounts for various performance ranges across different tasks, we first normalize the performance on each dataset according to $normalize(p) = \frac{p}{p_{\text{best}}} \times 100\%$, where p_{best} represents the best performance obtained in Table 1, then compute the average of normalized performance across all datasets, denoted by ANP. We report the best ANP using task-specific hyperparameters (ANP_{best}) and a fixed hyperparameter (ANP_{fix}) for each decoding method respectively. The results on Llama2-7B family are presented in Figure 2. For Llama2-7B, both FSD and FSD-d rank among the top-3 decoding methods in terms of performance, whether under task-specific hyperparameters (ANP_{best}) or one fixed hyperparameter (ANP_{fix}), demonstrating that these methods can have the ideal performance without the need for fine-grained selection of hyperparameters for each dataset. In contrast, while temperature sampling achieves comparable results in terms of ANP_{best}, it shows an 11.59% decrease in ANP_{fix} when hyperparameters are fixed, highlighting its sensitivity to hyperparameters. Similarly, for Llama2-7B-Chat, BS and DBS perform well and are not sensitive to hyperparameters, while temperature sampling still exhibits a 3.90% decrease. Notably, CD is also sensitive to hyperparameters, with a performance decrease of 9.42% on Llama2-7B and 3.35% on Llama2-7B-Chat.

4.3 Decoding Speed

We assess and compare the decoding speed of various decoding methods in Figure 3. For a more intuitive understanding, we calculate the latency ratio for each decoding method by normalizing their latency with respect to the latency of greedy search. To demonstrate how their latency grows with generation lengths, we plot the latency for generating 128, 256, 512 and 1024 tokens given 32 tokens using Llama2-7B. It is worth noting that we omit the results of all stochastic decoding methods mentioned in §2.2 because they achieve very close latency to that of greedy search. It is reasonable because their sampling processes only require negligible additional computation.

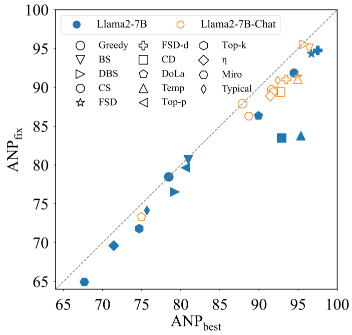

Figure 2: Hyperparameter Sensitivity. ANP_{best} and the best ANP_{fix} for each decoding method on Llama2-7B with blue solid markers ● and Llama2-7B-Chat with orange hollow markers ○ . The ANP_{fix} with the optimal hyperparameters for each decoding method are detailed in Appendix E.

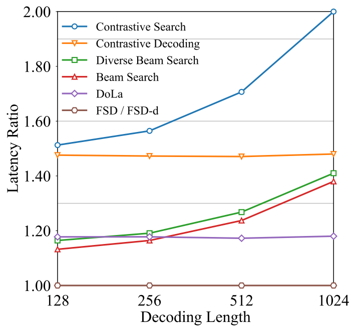

Figure 3: Decoding latency ratios. The latency is measured on one A6000 GPU with batch size = 1.

It can be observed that contrastive search is the decoding method with the slowest decoding speed. Moreover, the latency ratio grows considerably as generation length increases (from 1.51x to 2.00x slower than greedy search). This is due to that the look-ahead mechanism in contrastive search is very time-consuming. Contrastive decoding is about 1.4x slower than greedy search for the additional run of a smaller amateur model. However, the latency ratio of contrastive decoding remains constant across different lengths, indicating better adaptability for long sequence generation. Beam search and diverse beam search are faster than contrastive search and contrastive decoding but slower (1.13x to 1.41x) than greedy search. Both have latency ratios that grow approximately linearly with the sequence length while diverse beam search is slightly slower than beam search. The speed of DoLa is comparable to beam search and diverse beam search when the generation is relatively short (128 and 256). Nevertheless, their difference in-

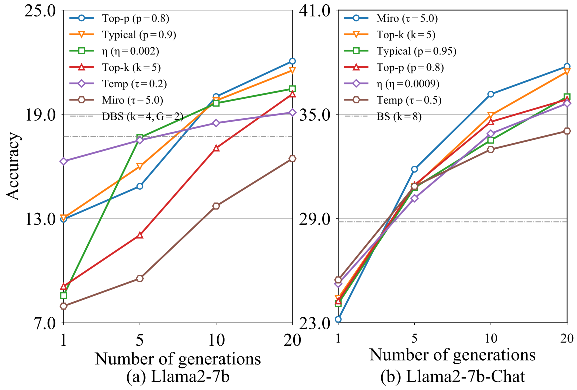

Figure 4: Results of stochastic decoding methods with self-consistency on GSM8K.

Model	Temp	Top-p	Top-k	η	Miro	Typical
7B	21.91 (0.7)	22.06 (0.8)	20.17 (5)	21.23 (0.004)	16.98 (4.0)	22.06 (0.95)
7B-Chat	36.92 (0.9)	36.85 (1.0)	37.68 (10)	35.63 (0.0009)	37.76 (5.0)	36.92 (0.90)

Table 3: Best results of different stochastic methods with self-consistency (20 generations) setting on GSM8K for Llama2-7B family. The best hyperparameters are annotated in parentheses.

creases as the generation length grows because the latency ratio of DoLa remains consistent across different lengths. Notably, FSD and FSD-d not only run as fast as greedy search but also maintain a consistent latency ratio across different lengths, underscoring their superior efficiency against other advanced deterministic decoding methods.

5 Further Analysis

5.1 Self-Consistency

Previous experiments demonstrate that the best-performing decoding methods are generally deterministic ones on closed-ended tasks, particularly on complex reasoning tasks such as the GSM8K dataset. Nonetheless, one unique advantage of stochastic decoding methods is that they can produce varied results through multiple runs, of which one can use the self-consistency strategy (Wang et al., 2023) for enhanced task performance. Concretely, self-consistency samples multiple generations and takes a majority vote to determine the final answer. To gain further insights into the potential of stochastic decoding methods, we then delve into the experiments with self-consistency.

As illustrated in Figure 4, we plot the accuracies of various stochastic decoding methods on GSM8K with respect to varying numbers of sampled generations (1, 5, 10, and 20). We also contrast the results with the best accuracies achieved by deterministic decoding methods (i.e., 17.74% by diverse beam search using Llama2-7B and 28.81% by beam search using Llama2-7B-Chat), denoted by the gray dashed lines. The results show that sampling a larger number of generations consistently leads to better performance, confirming the usefulness of self-consistency in taking advantage of the diversity introduced by stochastic sampling. Except for the results of mirostat sampling on Llama2-7B, we can see that all stochastic methods eventually surpass the best-performing deterministic methods when the number of sampling reaches 20. Note that the results in Figure 4 are obtained by using the best hyperparameters we find in Table 1 where only one-pass generation is allowed.

We speculate that further tuning the hyperparameters can improve the performance under the self-consistency strategy. Thus we undertake an additional hyperparameter search in scenarios where the number of generations is set to 20. The highest results along with the corresponding hyperparameters are reported in Table 3. Compared to the results in Figure 4, we can see that the performance is boosted by employing a hyperparameter with greater randomness or candidate pool. For example, on Llama2-7B-Chat, the accuracy of temperature sampling increases from 34.04% ($\tau = 0.5$) to 36.92% ($\tau = 0.9$). Another interesting finding is that the best hyperparameters for aligned models typically suggest greater randomness (e.g., $\tau = 0.9$ vs. $\tau = 0.7$ for temperature sampling).

5.2 Scaling Model Size

In order to investigate the impact of model scale on different decoding methods, we provide further experiments on Llama2 family with 13B and 70B parameters in 3 representative tasks: MBPP, GSM8K, and Wikinews. We present the results in Table 4. It can be observed that as the model's parameters increase, the *relative deviation percentage* (RDP) of each task decreases, indicating that the differences between different decoding methods have been reduced. This suggests that scaling model size can diminish the significance of decoding strategies. Moreover, as the number of model parameters varies, the optimal hyperparameters for each decoding method are also subject to change (detailed in Appendix E). Consequently, there is also a need to adjust the hyperparameters for larger-scale models individually, rather than directly applying those from smaller models. Meanwhile, the degree of impact from the model scale varies for different

Model	Dataset	Deterministic Methods								Stochastic Methods						RDP
		Greedy	BS	DBS	CS	FSD	FSD-d	CD	DoLa	Temp	Top-p	Top-k	η	Miro	Typical	
7B	MBPP	17.80	19.40	18.40	17.40	19.20	21.20	18.20	18.40	17.20	14.80	10.20	9.40	7.80	12.00	25.81
	GSM8K	13.87	17.21	17.74	14.63	16.83	16.60	17.21	15.39	16.30	12.96	9.10	8.64	7.96	13.04	23.06
	Wikinews	40.10	41.33	32.02	96.66	96.42	98.40	85.17	94.44	95.40	95.19	96.47	97.48	98.51	97.67	28.83
13B	MBPP	23.00	24.00	23.20	24.40	23.00	25.80	23.00	23.80	23.40	17.40	13.40	21.60	10.00	17.20	21.37
	GSM8K	28.81	29.64	29.19	29.42	31.99	31.16	33.36	28.58	30.02	24.94	18.20	30.10	15.39	21.76	18.74
	Wikinews	62.02	50.30	51.00	98.22	97.01	93.26	94.83	91.53	96.88	97.77	97.81	97.19	96.94	96.87	19.95
70B	MBPP	41.80	43.40	41.00	39.40	41.20	41.20	42.20	37.00	41.80	33.20	25.80	38.80	24.80	42.20	15.23
	GSM8K	57.39	59.44	58.76	58.91	60.73	60.42	63.91	61.33	57.47	53.37	44.20	58.53	38.36	59.89	11.92
	Wikinews	42.44	76.35	77.3 3	95.22	95.68	93.29	95.3	94.31	94.09	92.75	93.39	96.04	96.02	92.33	16.02
7B-Chat	MBPP	17.20	21.60	21.20	17.40	17.80	17.80	17.40	18.00	20.00	17.60	16.00	17.00	16.00	18.00	9.08
	GSM8K	24.79	28.81	26.91	25.70	25.40	24.56	26.46	22.14	25.47	24.26	24.41	25.25	23.20	24.11	6.25
	Wikinews	58.34	71.01	74.13	70.42	76.74	81.84	74.33	63.99	83.84	76.76	79.65	72.24	70.02	72.32	8.84
13B-Chat	MBPP	22.60	24.80	24.40	23.80	24.00	23.40	23.80	23.60	24.80	24.00	24.00	24.20	22.60	23.60	2.69
	GSM8K	34.57	39.73	38.06	36.24	36.62	36.16	36.62	33.13	36.32	36.01	35.41	35.41	36.01	36.85	4.05
	Wikinews	77.35	84.43	88.82	87.80	92.89	82.58	98.06	70.68	84.54	87.50	82.20	89.20	89.23	90.13	7.50
70B-Chat	MBPP	31.40	31.80	32.00	30.40	30.80	30.80	30.60	30.20	32.00	30.80	28.40	31.60	28.20	31.60	3.74
	GSM8K	51.93	50.87	53.90	53.22	52.01	52.54	52.24	48.82	52.62	52.99	51.10	52.92	51.93	52.16	2.28
	Wikinews	77.53	74.01	84.10	85.85	84.60	87.13	81.54	69.58	80.67	82.00	83.85	82.53	75.11	84.69	6.04

Table 4: Results of Llama2 family models with different scales on MBPP, GSM8K, Wikinews datasets. We report the relative deviation percentage (RDP) of the performance of different decoding methods on each task in the last column. The corresponding hyperparameters for each decoding method are listed in Appendix E.

Model	Dataset	Deterministic Methods								Stochastic Methods						RDP
		Greedy	BS	DBS	CS	FSD	FSD-d	CD	DoLa	Temp	Top-p	Top-k	η	Miro	Typical	
13B-INT4	MBPP	23.00	24.60	23.00	21.20	24.60	25.20	23.00	23.00	24.00	18.40	11.40	21.00	10.40	22.60	21.28
	GSM8K	27.45	30.33	28.13	27.67	31.01	30.10	31.46	29.11	27.60	21.30	15.69	26.38	14.10	27.90	19.92
	Wikinews	47.41	46.61	46.70	91.21	96.11	95.91	87.51	92.83	97.79	97.99	96.70	90.32	95.67	91.02	23.28
13B-INT8	MBPP	21.60	23.20	22.20	23.20	22.60	25.20	22.80	24.00	23.00	17.60	12.60	11.80	9.60	15.40	25.46
	GSM8K	28.43	28.89	28.96	29.04	30.93	30.48	33.59	28.28	29.34	23.88	17.21	16.45	13.34	21.68	23.25
	Wikinews	49.24	51.92	45.56	94.39	96.99	97.18	93.96	94.06	94.81	97.71	96.33	97.28	95.60	97.15	22.62
13B-Chat-INT4	MBPP	23.80	25.60	25.80	25.40	24.80	24.60	24.40	22.80	24.80	24.40	21.60	24.00	22.40	25.40	4.97
	GSM8K	34.12	35.71	37.45	34.50	35.33	35.41	34.42	31.61	35.33	34.27	33.97	34.04	33.74	35.33	3.64
	Wikinews	80.34	83.65	83.16	86.81	85.34	86.98	91.40	73.72	87.76	89.29	81.25	84.63	83.02	83.90	4.93
13B-Chat-INT8	MBPP	24.00	23.80	24.60	24.20	22.80	22.40	23.40	23.60	23.40	23.20	23.40	25.20	23.40	23.80	2.88
	GSM8K	35.56	36.92	37.68	37.76	36.69	36.69	37.38	31.54	36.09	38.44	37.15	36.92	36.85	37.38	4.29
	Wikinews	73.73	83.22	88.82	91.37	85.53	81.25	89.41	57.87	90.50	88.67	87.10	82.91	84.87	81.61	10.34

Table 5: Results for INT4 and INT8 quantization with Llama2 13B family on MBPP, GSM8K, Wikinews datasets. The corresponding hyperparameters for each decoding method are listed in Appendix E.

decoding methods. For example, in MBPP, the best performance of η sampling on Llama2-7B is 9.40%, which is less than half of the best method FSD-d at 21.20%. However, for Llama2-13B, η sampling achieves 21.60%, and for Llama2-70B, it reaches 38.80%, showing comparable efficacy to the best decoding method. This shows η sampling benefits greatly from a greater model scale.

5.3 Quantization

The large size of LLMs presents challenges for deployment, especially where resources are limited. Consequently, in the LLM era, it is crucial to examine how various decoding methods perform in quantization settings. We assess the performance of decoding methods in both INT8 quantization (Dettmers et al., 2022) and INT4 quantization (Lin et al., 2023b) for Llama2-13B family. As detailed in Table 5, compared with the FP16 13B model in Table 4, the RDP under quantized models is larger, indicating that quantization may impact the models' robustness to different decoding methods. At the same time, different decoding methods exhibit varying adaptability to quantized models. Specially, the performance changes of deterministic methods before and after quantization are not significant for both INT4 and INT8. However, for η and typical sampling, there are noticeable changes when quantizing Llama2-13B. Taking GSM8K as an example, η sampling under INT8 quantization decreased by 13.65%, while typical sampling under INT4 quantization improved by 6.14% on GSM8K. Typical and η sampling are more influenced by quantization because their computing involves numerically unstable calculations. This may indicate that the impact of quantization on the information entropy of different tokens during decoding cannot be ignored.

6 Conclusion

This study offered a comprehensive analysis of diverse traditional and contemporary decoding methods in the context of LLMs. Our experiments shed light on the efficacy, robustness, efficiency, and universality of these decoding methods across a range of tasks, models, and settings. One primary finding is that the choice of decoding methods remains crucial and different decoding methods manifests different advantages in different scenarios. However we still provide some practical guidelines in Appendix G. We hope this investigation provides valuable insights and guidance for practitioners and researchers in selecting and advancing decoding methods for LLMs.

Limitations

Despite the thoroughness of our study, there are some inherent limitations. First, while we have explored a variety of tasks and models, the ever-evolving nature of LLMs implies that new models or tasks might display distinct behaviors. Second, although our analysis of hyperparameter sensitivity covers a wide range of commonly used configurations, it is not exhaustive and does not account for all possible hyperparameters. Lastly, this paper does not explore the integration of multiple decoding methods, such as combining temperature sampling with a repetition penalty mechanism.

References

AI@Meta. 2024. Llama 3 model card.

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *ArXiv preprint*, abs/2309.16609.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. 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*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023a. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023b. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

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.

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. *ArXiv preprint*, abs/2110.14168.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *ArXiv preprint*, abs/2208.07339.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *ArXiv preprint*, abs/2304.07327.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. 2023a. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023c. Alpacaeval: An automatic evaluator of instruction-following models.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023a. The unlocking spell on base llms: Rethinking alignment via in-context learning. *ArXiv preprint*, abs/2312.01552.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023b. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv*.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *ArXiv preprint*, abs/2305.14251.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 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*, abs/2309.09117.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *ArXiv preprint*, abs/2308.12950.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

MosaicML NLP Team. 2023. Introducing mpt-30b: Raising the bar for open-source foundation models. Accessed: 2023-06-22.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10.

Haoran Yang, Deng Cai, Huayang Li, Wei Bi, Wai Lam, and Shuming Shi. 2023. A frustratingly simple decoding method for neural text generation.

Xuanyu Zhang and Qing Yang. 2023. Self-qa: Unsupervised knowledge guided language model alignment. *ArXiv preprint*, abs/2305.11952.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*.

A Decoding Strategies

A.1 Deterministic Methods

Greedy Search is arguably the simplest decoding strategy. At each time step t, it selects the token with the highest probability predicted by the model from the whole vocabulary set $\mathcal{V}$. Mathematically, the chosen token y_t at time t is:

$$y_t = \arg\max_{y \in \mathcal{V}} P(y|\mathbf{x}, \mathbf{y}_{<t}) \tag{1}$$

where $\mathbf{x}$ is the original input and $\mathbf{y}_{<t}$ is the generated tokens until time $t-1$. One drawback of greedy search is that it does not consider the global sequence score and can get stuck in local optima. This is why beam search is devised.

Beam Search (Freitag and Al-Onaizan, 2017) maintains a set, or "beam", of the k most probable sequences at each time step, where the hyperparameter k is referred to as the beam width. At time t, for each $\mathbf{y}_{<t} \in \mathcal{B}_{t-1}$, where $\mathcal{B}_{t-1}$ is the set of k most probable sequences at time $t-1$, it calculates a score for each token $y \in \mathcal{V}$:

$$\text{score}(\mathbf{y}_{<t}, y) = \log P(\mathbf{y}_{<t}, y|\mathbf{x}) \tag{2}$$

Then, a new set $\mathcal{B}_t$ is obtained:

$$\mathcal{B}_t = \underset{\mathbf{y}_{<t} \in \mathcal{B}_{t-1}, y \in \mathcal{V}}{\text{argtopk}} \ \text{score}(\mathbf{y}_{<t}, y) \tag{3}$$

We specifically test beam size 4 and 8 in our experiment.

Diverse Beam Search (Vijayakumar et al., 2018) is a variant of beam search and aims to improve the diversity among the generated sequences. It divides the k sequences into G groups, each with a size of k/G sequences. The algorithm operates in a similar way to the standard beam search, but instead of choosing the top-k sequences from all candidate sequences, it selects the top- k/G sequences for each group. The key difference lies in how the scores are calculated. In diverse beam search, a penalty is added to the score of a sequence if a similar sequence has already been in other groups:

$$\begin{aligned} \text{score}(\mathbf{y}_{<t}, y)_{\mathbf{y}_{<t} \in \mathcal{B}^g_{t-1}} &= \log P(\mathbf{y}_{<t}, y|\mathbf{x}) \\ &- \lambda \sum_{g' < g} \Delta((\mathbf{y}_{<t}, y), \mathcal{B}^{g'}_t) \end{aligned} \tag{4}$$

where $\Delta((\mathbf{y}_{<t}, y), \mathcal{B}^{g'}_t)$ is a measure of similarity between $(\mathbf{y}_{<t}, y)$ and sequences within $\mathcal{B}^{g'}_t$. In our experimental setup, we configure various (k, G) pairs of (4,2), (4,4), (8,2), (8,4), and the diversity penalty λ is always set to 1.

Contrastive Search (Su et al., 2022) assumes the LM has an isotropic representation space and adds a penalty term that decreases the generation probabilities of tokens producing hidden states that are very similar to the previous context. Formally, given the context $(\mathbf{x}, \mathbf{y}_{<t})$, the selection of the output y_t follows

$$\begin{aligned} y_t = \arg\max_{y \in V^k} & (1-\alpha) P(y|\mathbf{x}, \mathbf{y}_{<t}) \\ & -\alpha \max\{s(h_y, h_v) : v \in (\mathbf{x}, \mathbf{y}_{<t})\} \end{aligned} \tag{5}$$

where $\mathcal{V}^k$ is the set of top-k predictions from the language model's probability distribution $P(y|\mathbf{x}, \mathbf{y}_{<t})$. h_v is the hidden states for the token v, and s is the similarity function where the cosine similarity is usually adopted. We search α from $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$ in our experiment.

Contrastive Decoding (Li et al., 2023b) employs an additional amateur LM and penalizes undesired attributes associated with the amateur model. Formally, for each candidate token $y \in \mathcal{V}^c$

$$\text{score}((\mathbf{x}, \mathbf{y}_{<t}), y) = (1+\beta) * \mathbf{u}_y - \beta * \mathbf{v}_y \tag{6}$$

$\mathbf{u}$ and $\mathbf{v}$ are the logits before softmax of the expert and amateur models respectively. These two models have the same tokenizer and the expert model is usually much larger than the amateur model. $\mathcal{V}^c$ is a set of candidate tokens selected based on the following criteria:

$$\begin{aligned} &\mathcal{V}^c(\mathbf{x}, \mathbf{y}_{<t}) = \\ &\{y \in \mathcal{V} : P_{exp}(y|\mathbf{x}, \mathbf{y}_{<t}) > \alpha \max P_{exp}(\cdot|\mathbf{x}, \mathbf{y}_{<t})\} \end{aligned} \tag{7}$$

In our experiment, we adopt TinyLlama-1.1B[2] as the amateur model. We use the default setting with α set to 0.1 and we search β from $[0.1, 0.3, 0.5, 0.7, 0.9]$.

Frustratingly Simple Decoding (Yang et al., 2023) exploits the contrasts between the LLM and an auxiliary anti-LM constructed based on the current prefix. There are two variants of FSD: FSD and FSD-d depending on whether the anti-LM is implemented as a vectorized or discrete n-gram model. Specifically, the FSD score is defined as

[2] https://huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-955k-token-2T

$$\mathrm{FSD}(y|\mathbf{x}, \mathbf{y}_{<t}) = (1-\alpha) P_\theta(y|\mathbf{x}, \mathbf{y}_{<t}) - \alpha \times P_\omega(y|\mathbf{x}, \mathbf{y}_{<t}) \tag{8}$$

where P_θ and P_ω represent the LM and the anti-LM respectively. The hyper-parameter $\alpha \geq 0$ is used to balance the two scores. In practice, it first selects the top-k most probable tokens according to $P_\theta(\cdot|\mathbf{x}, \mathbf{y}_{<t})$, denoted by $\mathcal{V}^k$. The token in $\mathcal{V}^{(k)}$ with the largest FSD score is chosen as the t_{th} token. We search α from $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$.

DoLa (Chuang et al., 2023) obtains the next-token distribution by contrasting the logits differences between the last layer and a premature layer. For Llama2-7b, the premature layer is dynamically selected from even-numbered layers from $[0, 16)$ and $[16, 32)$. For Llama2-13b, the ranges are $[0, 20)$ and $[20, 40)$. For Llama2-70b, the ranges are $[0, 20)$ and $[60, 80)$. They adopt the Jensen-Shannon divergence (JSD) as the measure of distance between the next-word distributions and select the layer that has the largest JSD as the premature layer.

A.2 Stochastic Methods

Temperature Sampling is a decoding strategy to control the randomness in the sampling process. Instead of directly sampling tokens from the predicted distribution, temperature sampling introduces a hyperparameter "temperature" τ that is used to adjust the probability distribution:

$$P(y|\mathbf{x}, \mathbf{y}_{<t}) = \frac{\exp(\mathbf{u}_y/\tau)}{\sum_j \exp(\mathbf{u}_j/\tau)} \tag{9}$$

where $\mathbf{u}_y$ is the logit of y before softmax. We conduct our experiment for τ within the range of 0.1 to 0.9, incrementing in value of 0.1.

Top-k Sampling (Fan et al., 2018) is used to ensure that the less probable words, which are in the unreliable tail of the distribution (Holtzman et al., 2020), should not have any chance to be selected. Only top-k probable tokens are considered for a generation. we explore a range of k values, specifically $[5, 10, 20, 50, 100]$.

Top-p Sampling (Holtzman et al., 2020) considers the minimal set of top tokens $\mathcal{V}^p$ that cover a specified percentage p of the distribution:

$$\sum_{y \in \mathcal{V}^p} P(y|\mathbf{x}, \mathbf{y}_{<t}) \geq p \tag{10}$$

For our study, we have examined various p thresholds, specifically $[0.8, 0.85, 0.9, 0.95, 1]$.

Typical Sampling (Meister et al., 2023) sorts the vocabulary according to the differences between distribution entropy and probabilities. The authors argue that the desired sequences should have information content close to the expected information content, i.e., the conditional entropy of the model. The candidate set $\mathcal{V}^c$ is a solution of the following problem:

$$\min_{\mathcal{V}^c} \sum_{y \in \mathcal{V}^c} |H(Y_t|\mathbf{x}, \mathbf{y}_{<t}) + \log P(y|\mathbf{x}, \mathbf{y}_{<t})| \quad \text{s.t.} \sum_{y \in \mathcal{V}^c} P(y|\mathbf{x}, \mathbf{y}_{<t}) \geq p \tag{11}$$

In our experiments, we vary the threshold p across the values $[0.2, 0.9, 0.92, 0.95]$ to examine its effect on sequence generation.

Top-η Sampling (Hewitt et al., 2022) truncates words whose probabilities are below an entropy-dependent threshold. The candidate set $\mathcal{V}^c$ is determined by:

$$\mathcal{V}^c = \{y \in \mathcal{V} | P(y|\mathbf{x}, \mathbf{y}_{<t}) \geq \sqrt{\eta} \exp(-h_{\theta,(\mathbf{x},\mathbf{y}_{<t})})\} \tag{12}$$

where $h_{\theta,(\mathbf{x},\mathbf{y}_{<t})}$ is the entropy of $P(Y|\mathbf{x}, \mathbf{y}_{<t})$. η is searched from $[0.0003, 0.0006, 0.0009, 0.002, 0.004]$.

Mirostat Sampling (Basu et al., 2021) directly control the perplexity rate of the generated text. It firstly estimates the value of s assuming words follow Zipf's law where s is an exponent characterizing the distribution. Then it uses top-k sampling to generate the new token where k is a function of the estimated s and of the target perplexity τ of the output text. We search τ from $[2.5, 3, 4, 5]$.

B Evaluation Benchmarks

B.1 Coding

HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021) are extensively utilized benchmarks within the measurement of LLM's code generating ability. These benchmarks encompass a vast collection of Python programming problems.

HumanEval (Chen et al., 2021) consists of 164 original programming problems by giving docstrings to generate code, which has an average of

9.6 test cases allocated to each problem. We use 0-shot prompt for both unaligned and aligned models.

MBPP (Austin et al., 2021) focus on generating code based on textual descriptions, which offers a set of 500 test programming problems, accompanied by three automated test cases per problem. We use 0-shot prompt for aligned models and 3-shot prompt for unaligned models.

B.2 Math Problem Soving

We utilize GSM8K (Cobbe et al., 2021) for assessing reasoning and problem-solving proficiencies within the domain of mathematics.

GSM8K (Cobbe et al., 2021) collects 1,319 high-quality linguistically diverse grade school math word problems as the test set, and reports 8-shot pass@1 accuracy. We use 0-shot prompt for aligned models and 8-shot prompt for unaligned models.

B.3 Summarization

We select the CNN/DailyMail (Hermann et al., 2015) and XSUM (Narayan et al., 2018) datasets, which are the most well-studied datasets in the literature on summarization faithfulness. This also ensures domain coverage of news-type data. Importantly, these datasets differ along a central axis studied in summarization:

XSUM (Narayan et al., 2018) is a dataset with largely abstractive reference summaries (meaning the string overlap between the document and its summary in the dataset is relatively small on average) which feature articles from the British Broadcasting Corporation (BBC). The test splits for the dataset are 11.5K examples. We use 0-shot prompt for aligned models and 1-shot prompt for unaligned models.

CNN/DailyMail (Hermann et al., 2015) is a dataset with largely extractive reference summaries that contain news articles from CNN and the DailyMail along with highlights that act as a summary for the article. The test splits for the dataset are 11.3K examples. We use 0-shot prompt for aligned models and 1-shot prompt for unaligned models. The model-generated summary is compared against a human-authored reference summary using automated metrics for overall quality ROUGE-L (Lin, 2004). Note that we randomly select 1,000 cases each from CNNDailyMail and XSUM for evaluation.

B.4 Translation

We evaluate the translation performance on WMT22 (Bojar et al., 2017) test sets.

WMT22 Competition (Bojar et al., 2017) constructed based on more recent content from various domains, including news, social, e-commerce, and conversational domains. The numbers of samples for De $\Rightarrow$ En, En $\Rightarrow$ De, Zh $\Rightarrow$ En and En $\Rightarrow$ Zh tasks are 1984, 2037, 1875 and 2037, respectively. For automatic evaluation, we adopt BLEU (Papineni et al., 2002) implementated in SacreBLEU (Post, 2018)[3]. We use 3-shot prompt for unaligned models.

B.5 Commonsense reasoning

Commonsense reasoning is key for interacting with the world and is still beyond the reach of current natural language understanding systems (Talmor et al., 2019). We consider measuring open-ended performance on two datasets covering a diverse range of commonsense reasoning types from BIG-Bench (Srivastava et al., 2022), CommonsenseQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021).

CommonsenseQA (Talmor et al., 2019) asks commonsense questions about the world involving complex semantics that often require prior knowledge. There are a total of 1.22k instances in the CommonsenseQA validation set. We use 6-shot prompt for aligned models and 1-shot prompt for unaligned models.

StrategyQA (Geva et al., 2021) requires models to infer a multi-hop strategy to answer questions. We use the open-domain setting (question-only set) from BIG-Bench (Srivastava et al., 2022) which contains 2.29k test instances. We use 0-shot prompt for aligned models and 4-shot prompt for unaligned models. The two BIG-bench tasks do not have training sets, so we select the first ten examples as exemplars in the evaluation set as few-shot exemplars and report accuracy on the rest of the evaluation set.

B.6 Factual Knowledge

Factual Knowledge refers to their tendency to generate factual errors. This is considered a critical

[3] https://github.com/mjpost/sacrebleu

issue in LLMs because it is challenging for users to identify and poses real-life risks.

FActScore (Min et al., 2023) scrutinizes the factual accuracy of biographies generated by LLMs for 500 specific individuals. Conducting a pipeline to transform a long-form model generation into pieces of atomic statements and measure the atomic statement's accuracy with retrieved knowledge. We use 0-shot prompt for aligned models.

B.7 Instruction Following

For our research, we select the representative broad-coverage benchmark Alpace-eval (Li et al., 2023c).

Alpace-eval (Li et al., 2023c) assess the LLM's generation quality by 805 prompts from several sources: Vicuna (Chiang et al., 2023b) (80 prompts), Self-instruct (Zhang and Yang, 2023) (252 prompts), Open Assistant (Köpf et al., 2023) (188 prompts), Koala (Geng et al., 2023) (156 prompts), HH_RLHF (Bai et al., 2022) (129 prompts), quantifying the pairwise Win Rate against a reference model, Text-Davinci-003.

B.8 Open-ended Text Generation

Open-ended text generation aims to craft fluent and coherent textual continuations of given prompts. Following (Li et al., 2023b), we evaluate three domains for open-ended text generation: Book, Wikinews, Wikitext.

Book contains 1,947 prompts collected from BookCorpus (Zhu et al., 2015)for story generations. We use 0-shot prompt for both unaligned and aligned models.

Wikinews include 2,000 news articles prompts collected from Wikinews[4]. We use 0-shot prompt for both unaligned and aligned models.

Wikitext select 1,314 prompts from wikitext-103 (Merity et al., 2017) as the Wikipedia representative domain. We use 0-shot prompt for both unaligned and aligned models. We utilize MAUVE (Pillutla et al., 2021) score (the higher the better) to measure the distribution similarity between the set of generated text and the set of gold references. Note that we randomly select 500 cases each from among the three domains mentioned above for evaluation.

[4]http://www.wikinews.org

C Instruction Template

The instruction templates for each dataset are list from Table 6 to Table 25.

D Different Foundation Models

We extend our analysis to investigate the decoding methods under different foundation models[5]. We select several representive models, including CodeLlama-7b (Rozière et al., 2023), Qwen-7B (Bai et al., 2023), MPT-7b (Team, 2023), Mistral-7B (Jiang et al., 2023), deepseek-moe-16b-base (Dai et al., 2024), Llama-3-8B (AI@Meta, 2024) and their aligned versions. Addtionally we select vicuna-7b-v1.5 (Chiang et al., 2023a) which is an SFT-ed model from llama2-7B without RHLF for analysis. It is crucial to underscore that these models vary significantly in several aspects, such as pre-training data, model architecture, and etc. As illustrated in Tables 26, the results observed in §4.1 are still applicable to LLMs with different architectures. Detailed as below: i) For unaligned models, deterministic methods generally perform better than stochastic methods on all tasks except open-ended text generation. ii) Aligned models are less dependent on decoding methods than unaligned models. iii) Among stochastic methods, temperature sampling generally performs better, particularly when using unaligned models. Apart from these consistencies, it is worth noting that different decoding methods may result in different performance rankings for LLMs. For instance, Codellama outperforms Qwen by 7.60% in the MBPP with top-k sampling, yet lags behind by 2.20% with η sampling. This implies that different models still have varying adaptability to specific decoding methods, suggesting that the selection of decoding strategies should be more meticulously rigorous during the evaluation of LLMs.

E Settings of Hyperparameters

The optimal hyperparameters for each decoding method across different datasets and models are listed from Table 27 to Table 31.

F Ouput of DoLa

The output examples that DoLa fails to terminate its generation appropriately are listed in Table 32.

[5] CD and DoLa are not included. Because it is challenging to find an amateur model for each foundation model for CD, and for DoLa, it is difficult to determine the appropriate number of layers for logits comparison for individual models.

PROMPT FOR HUMANEVAL
[DOCSTRING]

Table 6: 0-shot prompt for HumanEval (unaligned model).

PROMPT FOR HUMANEVAL
Please complete the remaining Python function code based on the following docstring content.
[DOCSTRING]

Table 7: 0-shot prompt for HumanEval (aligned model).

G Practical Guidelines

Our study underscores the significance of selecting an appropriate decoding method in the era of large language models (LLMs). Despite the advancements in LLMs, our findings indicate that the choice of decoding method remains critical and cannot be overlooked. This decision is contingent upon several factors, including the specific test task, the model being used, and the priority—whether it is performance, robustness, or speed. The core contribution of our paper lies in demonstrating the nuanced and complex nature of decoding method selection. The optimal decoding strategy is not universally applicable and varies based on the aforementioned factors. This complexity underscores the necessity for a comprehensive evaluation framework in future research and highlights the need for practitioners to consider multiple dimensions when deploying LLMs. Despite the intricacies involved, we offer several practical guidelines for deploying LLMs without extensive hyperparameter searching: For quick setup, Unaligned Models (e.g., Llama2-7B): For these models, we recommend using either FSD or FSD-d; Aligned Models (e.g., Llama2-7B-Chat): BS or DBS is advised for aligned models to achieve satisfactory performance. When computational resources allow for self-consistency: Unaligned Models (e.g., Llama2-7B): implementing temperature sampling with an optimal temperature setting of 0.7 is recommended to enhance model performance. Aligned Models (e.g., Llama2-7B-Chat): a higher optimal temperature of 0.9 is suggested.

PROMPT FOR MBPP

You are an expert Python programmer, and here is your task: Write a function to find the similar elements from the given two tuple lists. Your code should pass these tests:

```
assert similar_elements((3, 4, 5, 6),(5, 7, 4, 10)) == (4, 5)
assert similar_elements((1, 2, 3, 4),(5, 4, 3, 7)) == (3, 4)
assert similar_elements((11, 12, 14, 13),(17, 15, 14, 13)) == (13, 14)
[BEGIN]
def similar_elements(test_tup1, test_tup2):
    res = tuple(set(test_tup1) & set(test_tup2))
    return (res)
[DONE]
```

You are an expert Python programmer, and here is your task: Write a python function to identify non-prime numbers. Your code should pass these tests:

```
assert is_not_prime(2) == False
assert is_not_prime(10) == True
assert is_not_prime(35) == True
[BEGIN]
import math
def is_not_prime(n):
    result = False
    for i in range(2,int(math.sqrt(n)) + 1):
        if n % i == 0:
            result = True
    return result
[DONE]
```

You are an expert Python programmer, and here is your task: Write a function to find squares of individual elements in a list using lambda function. Your code should pass these tests:

```
assert square_nums([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])==[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
assert square_nums([10,20,30])==([100,400,900])
assert square_nums([12,15])==([144,225])
[BEGIN]
def square_nums(nums):
    square_nums = list(map(lambda x: x ** 2, nums))
    return square_nums
[DONE]
```

You are an expert Python programmer, and here is your task: [TASK_DEFINATION]. Your code should pass these tests:

```
[TEST_CASE_1]
[TEST_CASE_2]
[TEST_CASE_2]
[BEGIN]
```

Table 8: 3-shot promp for MBPP (unaligned model).

PROMPT FOR MBPP

You are an expert Python programmer, and here is your task: [TASK_DEFINATION]. Your code should pass these tests:

```
[TEST_CASE_1]
[TEST_CASE_2]
[TEST_CASE_2]
```

Table 9: 0-shot promp for MBPP (aligned model).

PROMPT FOR GSM8K

Question: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

Answer: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Question: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

Answer: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5.

Question: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?

Answer: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39. The answer is 39.

Question: Question: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?

Answer: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8. The answer is 8.

Question: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?

Answer: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9. The answer is 9.

Question: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

Answer: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29.

Question: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?

Answer: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls. The answer is 33.

Question: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?

Answer: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8.

Question: `[QUESTION]`

Answer:

Table 10: 8-shot prompt for GSM8K (unaligned model).

PROMPT FOR GSM8K

Please answer the math questions below.
`[QUESTION]`
You need to first take step-by-step reasoning and then give the final result.

Table 11: 0-shot prompt for GSM8K (aligned model).

PROMPT FOR XSUM
Article: The Bath-born player, 28, has made 36 appearances for the Dragons since joining from Wasps in 2015. He is in his second season and signed a contract extension in December 2016. Dragons forwards coach Ceri Jones said: "It's a big blow. Eddie has been excellent all year for us, he has really stepped up to the mark and will be a big loss." However, Jones says Jackson's misfortune can be a chance for others to thrive. "We are very fortunate to have the likes of Ollie Griffiths, Harrison Keddie, James Thomas who can come into the back-row," said Jackson. "Harri has shown glimpses of what he can do all season and there's definitely a player there, so this is an opportunity." Dragons travel to Munster in the Pro12 on Friday.

Summarize the above article in 1 sentence.
Newport Gwent Dragons number eight Ed Jackson has undergone shoulder surgery and faces a spell on the sidelines. **Article:**

`[ARTICLE]`
SSummarize the above article in 1 sentence.

Table 12: 1-shot prompt for XSUM (unaligned model).

PROMPT FOR XSUM
Article: `[ARTICLE]`

Summarize the above article in 1 sentence.

Table 13: 0-shot prompt for XSUM (aligned model).

PROMPT FOR CNNDAILYMAIL
Article: PARIS, France (CNN) – Interpol on Monday took the unprecendented step of making a global appeal for help to identify a man from digitally reconstructed photos taken from the Internet that it said showed him sexually abusing underage boys. This moving image shows how police used software to unscramble the image. (Source: Interpol) The man's face was disguised by digital alteration, but the images were capable of being restored, according to a bulletin from Interpol – the international police agency based in Lyon, France. Interpol Secretary General Ronald K. Noble said the pictures have been on the the Internet for several years, but investigators have been unable to determine the man's identity or nationality. "We have tried all other means to identify and to bring him to justice, but we are now convinced that without the public's help this sexual predator could continue to rape and sexually abuse young children whose ages appear to range from six to early teens," Noble said. He said there is "very good reason to believe that he travels the world in order to sexually abuse and exploit vulnerable children." Interpol has determined the photos were taken in Vietnam and Cambodia. "The decision to make public this man's picture was not one which was taken lightly," said Kristin Kvigne, assistant director of Interpol's Trafficking in Human Beings Unit. The suspect's photo and more information can be seen online at Interpol's Web site. E-mail to a friend .
Summarize the above article in 3 sentences.
Man posted photos on the Internet of himself sexually abusing underage boys . Computer experts managed to undo digital masking to reveal the man . Man abused 12 boys in Vietnam and Cambodia .

Article: `[ARTICLE]`
Summarize the above article in 3 sentences.

Table 14: 1-shot prompt for CNN/Dailymail (unaligned model).

PROMPT FOR CNNDAILYMAIL
Article: `[ARTICLE]`

Summarize the above article in 3 sentences.

Table 15: 0-shot prompt for CNN/Dailymail (aligned model).

PROMPT FOR WMT DE⇒EN Translate the following sentence from German to English. [GERMAN] Frau Schroedter, ich bin gerne bereit, die damit zusammenhängenden Fakten zu prüfen, wenn mir Ihr Brief vorliegt. [ENGLISH] Yes, Mrs Schroedter, I shall be pleased to look into the facts of this case when I have received your letter. Translate the following sentence from German to English. [GERMAN] Das ist der Fall von Alexander Nikitin. [ENGLISH] It is the case of Alexander Nikitin. Translate the following sentence from German to English. [GERMAN] Meine Frage betrifft eine Angelegenheit, die am Donnerstag zur Sprache kommen wird und auf die ich dann erneut verweisen werde. [ENGLISH] My question relates to something that will come up on Thursday and which I will then raise again. Translate the following sentence from German to English. [GERMAN] [GERMAN_TEXT] [ENGLISH]

Table 16: 3-shot prompt for WMT De⇒En (unaligned model).

PROMPT FOR WMT EN⇒DE Translate the following sentence from English to German. [ENGLISH] Yes, Mrs Schroedter, I shall be pleased to look into the facts of this case when I have received your letter. [GERMAN] Frau Schroedter, ich bin gerne bereit, die damit zusammenhängenden Fakten zu prüfen, wenn mir Ihr Brief vorliegt. Translate the following sentence from English to German. [ENGLISH] It is the case of Alexander Nikitin. [GERMAN] Das ist der Fall von Alexander Nikitin. Translate the following sentence from English to German. [GERMAN] Meine Frage betrifft eine Angelegenheit, die am Donnerstag zur Sprache kommen wird und auf die ich dann erneut verweisen werde. [ENGLISH] My question relates to something that will come up on Thursday and which I will then raise again. Translate the following sentence from English to German. [ENGLISH] [ENGLISH_TEXT] [GERMAN]

Table 17: 3-shot prompt for WMT En⇒De (unaligned model).

PROMPT FOR WMT ZH⇒EN
Translate the following sentence from Chinses to English.

[CHINESE] 柏林——2008年爆发的全球金融和经济危机是自大萧条以来最严峻的一次经济压力测试，也是自二战以来社会和政治制度所面临的最严重挑战。
[ENGLISH] BERLIN – The global financial and economic crisis that began in 2008 was the greatest economic stress-test since the Great Depression, and the greatest challenge to social and political systems since World War II.

Translate the following sentence from Chinses to English.

[CHINESE] 欧洲在避免债务和捍卫欧元的名义下正变得谨慎，而美国已经在许多方面行动起来，以利用这一理想的时机来实行急需的结构性改革。
[ENGLISH] Europe is being cautious in the name of avoiding debt and defending the euro, whereas the US has moved on many fronts in order not to waste an ideal opportunity to implement badly needed structural reforms.

Translate the following sentence from Chinses to English.

[CHINESE] 百年愚顽
[ENGLISH] One Hundred Years of Ineptitude

Translate the following sentence from Chinses to English.

[CHINESE] [CHINESE_TEXT]
[ENGLISH]

Table 18: 3-shot prompt for WMT Zh⇒En (unaligned model).

PROMPT FOR WMT EN⇒ZH
Translate the following sentence from English to Chinese.

[ENGLISH] BERLIN – The global financial and economic crisis that began in 2008 was the greatest economic stress-test since the Great Depression, and the greatest challenge to social and political systems since World War II.
[CHINESE] 柏林——2008年爆发的全球金融和经济危机是自大萧条以来最严峻的一次经济压力测试，也是自二战以来社会和政治制度所面临的最严重挑战。

Translate the following sentence from English to Chinese.

[ENGLISH] Europe is being cautious in the name of avoiding debt and defending the euro, whereas the US has moved on many fronts in order not to waste an ideal opportunity to implement badly needed structural reforms.
[CHINESE] 欧洲在避免债务和捍卫欧元的名义下正变得谨慎，而美国已经在许多方面行动起来，以利用这一理想的时机来实行急需的结构性改革。

Translate the following sentence from English to Chinese.

[ENGLISH] One Hundred Years of Ineptitude
[CHINESE] 百年愚顽

Translate the following sentence from English to Chinese.

[ENGLISH] [ENGLISH_TEXT]
[CHINESE]

Table 19: 3-shot prompt for WMT En⇒Zh (unaligned model).

PROMPT FOR COMMONSENSEQA

Question: What do people use to absorb extra ink from a fountain pen? Answer Choices: (a) shirt pocket (b) calligrapher's hand (c) inkwell (d) desk drawer (e) blotter

Answer: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink. So the answer is (e).

Question: What home entertainment equipment requires cable?
Answer Choices: (a) radio shack (b) substation (c) television (d) cabinet

Answer: The answer must require cable. Of the above choices, only television requires cable. So the answer is (c).

Question: The fox walked from the city into the forest, what was it looking for? Answer Choices: (a) pretty flowers (b) hen house (c) natural habitat (d) storybook

Answer: The answer must be something in the forest. Of the above choices, only natural habitat is in the forest. So the answer is (b).

Question: Sammy wanted to go to where the people were. Where might he go? Answer Choices: (a) populated areas (b) race track (c) desert (d) apartment (e) roadblock

Answer: The answer must be a place with a lot of people. Of the above choices, only populated areas have a lot of people. So the answer is (a).

Question: Where do you put your grapes just before checking out? Answer Choices: (a) mouth (b) grocery cart (c)super market (d) fruit basket (e) fruit market

Answer: The answer should be the place where grocery items are placed before checking out. Of the above choices, grocery cart makes the most sense for holding grocery items. So the answer is (b).

Question: Google Maps and other highway and street GPS services have replaced what? Answer Choices: (a) united states (b) mexico (c) countryside (d) atlas

Answer: The answer must be something that used to do what Google Maps and GPS services do, which is to give directions. Of the above choices, only atlases are used to give directions. So the answer is (d).
Question: `[QUESTION]`

Answer:

Table 20: 6-shot prompt for CommonsenseQA (unaligned model).

PROMPT FOR COMMONSENSEQA

Which choice is the correct answer to the question?
Question: `[QUESTION]`

Answer: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink. So the answer is (e).
Let's think step by step.

Table 21: 0-shot prompt for CommonsenseQA (aligned model).

PROMPT FOR STRATEGYQA

Question: Do hamsters provide food for any animals?

Answer: Hamsters are prey animals. Prey are food for predators. Thus, hamsters provide food for some animals. So the answer is yes.

Question: Could Brooke Shields succeed at University of Pennsylvania?

Answer: Brooke Shields went to Princeton University. Princeton University is about as academically rigorous as the University of Pennsylvania. Thus, Brooke Shields could also succeed at the University of Pennsylvania. So the answer is yes.

Question: Yes or no: Hydrogen's atomic number squared exceeds number of Spice Girls?

Answer: Hydrogen has an atomic number of 1. 1 squared is 1. There are 5 Spice Girls. Thus, Hydrogen's atomic number squared is less than 5. So the answer is no.

Question: Yes or no: Is it common to see frost during some college commencements?

Answer: College commencement ceremonies can happen in December, May, and June. December is in the winter, so there can be frost. Thus, there could be frost at some commencements. So the answer is yes.

Question: `[QUESTION]`

Answer:

Table 22: 4-shot prompt for StrategyQA (unaligned model).

PROMPT FOR COMMONSENSEQA

Which choice is the correct answer to the question?
Question: `[QUESTION]`

Answer: The answer must be an item that can absorb ink. Of the above choices, only blotters are used to absorb ink. So the answer is (e).
Let's think step by step.

Table 23: 0-shot prompt for StrategyQA (aligned model).

PROMPT FOR BOOK, WIKINEWS AND WIKITEXT

`[BEGIN_OF_TEXT]`

Table 24: 0-shot prompt for Book, Wikinews and Wikitext (unaligned model).

PROMPT FOR BOOK, WIKINEWS AND WIKITEXT

Please help me complete the text continuation based on the following content.
`[BEGIN_OF_TEXT]`

Table 25: 0-shot prompt for Book, Wikinews and Wikitext (aligned model).

Model	Dataset	Deterministic Methods						Stochastic Methods					
		Greedy	BS	DBS	CS	FSD	FSD-d	Temp	Top-p	Top-k	η	Miro	Typical
CodeLlama-7b	MBPP	35.40	34.20	35.00	36.00	37.00	39.60	35.00	32.80	25.40	23.60	21.20	31.80
	GSM8K	11.98	13.12	12.21	12.89	13.50	13.80	12.59	12.66	8.64	7.43	6.90	8.87
	Wikinews	10.49	9.81	8.85	87.99	97.19	94.35	90.89	94.63	96.69	97.59	96.34	94.06
CodeLlama-7b-Instruct	MBPP	36.80	40.80	41.60	37.00	37.20	36.60	39.00	37.60	35.60	35.40	34.40	38.20
	GSM8K	22.14	27.75	28.35	23.28	22.67	21.91	23.96	22.14	18.04	19.26	18.27	21.99
	Wikinews	90.11	96.75	85.83	90.39	92.87	92.21	87.69	89.26	90.39	90.31	94.32	84.99
Qwen-7B	MBPP	33.00	34.40	33.20	28.40	33.00	33.60	33.80	27.40	19.80	25.80	18.40	27.00
	GSM8K	53.22	57.32	56.56	50.04	53.53	54.59	53.90	47.46	38.67	49.05	36.24	51.86
	Wikinews	50.58	61.66	51.59	94.50	94.22	95.24	94.50	94.94	95.51	96.08	93.94	94.69
Qwen-7B-Chat	MBPP	30.40	30.80	33.60	25.80	30.80	29.80	30.00	28.80	26.80	24.20	25.00	27.20
	GSM8K	48.29	51.48	51.18	43.82	47.46	48.37	48.52	45.34	41.02	43.37	41.85	43.67
	Wikinews	73.43	89.12	90.75	91.87	89.40	88.07	89.85	88.11	85.43	84.37	81.37	80.04
MPT-7b	MBPP	18.20	22.80	21.00	21.20	21.40	21.80	19.00	14.80	11.20	8.40	6.60	11.40
	GSM8K	8.64	9.63	9.70	10.24	8.95	10.24	9.55	6.75	6.37	5.91	5.08	5.76
	Wikinews	22.44	6.08	7.30	87.85	97.96	97.58	96.89	96.19	98.56	96.95	97.83	97.80
MPT-7b-Instruct	MBPP	20.80	25.40	23.80	23.20	23.20	22.20	23.40	18.60	15.80	14.20	14.20	16.60
	GSM8K	4.93	2.88	4.09	2.88	6.75	5.91	6.75	4.85	5.46	2.65	4.09	2.58
	Wikinews	92.76	94.88	87.21	97.10	93.19	96.11	95.53	96.04	97.02	50.22	96.53	53.60
Mistral-7B	MBPP	36.40	41.80	41.60	39.60	39.20	38.60	37.40	32.60	28.00	24.20	21.80	28.40
	GSM8K	43.90	46.70	45.79	43.44	45.26	45.94	43.75	38.81	38.58	28.20	23.58	35.19
	Wikinews	46.74	45.81	35.44	93.88	92.58	89.67	88.60	94.70	91.64	92.78	94.03	93.34
Mistral-7B-Instruct	MBPP	29.00	28.20	27.20	27.40	27.60	27.40	28.80	28.40	27.40	26.60	25.80	26.40
	GSM8K	43.75	49.05	46.93	42.61	43.52	43.67	44.05	43.82	43.59	43.29	43.52	44.43
	Wikinews	79.89	82.40	83.97	76.86	85.96	83.47	88.36	89.45	83.78	84.85	82.54	90.61
deepseek-moe-16b-base	MBPP	35.20	36.20	35.80	34.20	36.60	36.80	28.20	26.60	19.00	18.80	21.20	28.20
	GSM8K	18.95	18.20	18.42	19.94	18.42	18.27	16.3	12.13	12.89	12.66	11.98	10.77
	Wikinews	41.16	42.80	40.66	94.79	95.42	96.74	96.30	97.42	98.01	96.83	96.16	95.69
deepseek-moe-16b-chat	MBPP	41.00	41.80	41.20	39.20	39.40	38.20	36.20	36.40	31.20	33.20	31.00	32.20
	GSM8K	50.11	50.64	48.90	50.95	46.25	47.61	39.50	45.41	36.77	31.69	27.14	36.09
	Wikinews	75.44	80.34	87.94	92.38	83.90	91.74	90.07	89.35	92.41	91.04	91.68	90.44
Llama-3-8B	MBPP	43.20	49.80	52.60	45.60	48.20	49.00	43.20	26.00	26.80	45.60	28.20	43.60
	GSM8K	48.45	51.18	46.55	48.60	51.25	52.46	46.47	23.65	25.09	44.96	26.16	46.78
	Wikinews	46.49	45.65	32.65	87.47	93.76	96.41	95.81	96.46	96.85	90.74	96.67	83.47
Llama-3-8B-Instruct	MBPP	49.60	50.00	49.00	46.80	48.60	49.20	49.00	48.40	45.80	48.20	45.60	48.80
	GSM8K	69.60	71.04	69.83	68.39	70.43	68.61	68.76	65.05	65.81	68.84	64.52	69.67
	Wikinews	54.13	51.81	75.72	81.81	79.41	75.28	68.23	71.36	37.08	72.35	50.53	76.55
vicuna-7b-v1.5	MBPP	22.60	24.60	25.80	21.60	22.60	22.40	23.40	21.20	17.80	21.60	19.20	22.40
	GSM8K	18.04	25.47	21.68	20.17	19.03	19.18	19.71	18.95	16.15	20.02	15.77	19.71
	Wikinews	84.63	94.43	90.19	90.70	91.37	80.70	89.47	87.65	85.33	85.02	89.58	89.22

Table 26: Results for different foundation models on MBPP, GSM8K, Wikinews datasets. The corresponding hyperparameters for each decoding method are listed in Table 30.

Model	Dataset	Deterministic Methods								Stochastic Methods					
		Greedy	BS	DBS	CS	FSD	FSD-d	CD	DoLa	Temp	Top-p	Top-k	η	Miro	Typical
Llama2-7B	HumanEval	-	8	8_2	0.4	0.2	0.4	0.3	[16,32)	0.4	0.8	20	0.0006	5.0	0.95
	MBPP	-	4	8_2	0.3	0.2	0.4	0.3	[16,32)	0.3	0.8	5	0.002	4.0	0.9
	GSM8K	-	8	4_2	0.4	0.3	0.4	0.3	[0, 16)	0.2	0.8	5	0.004	5.0	0.9
	XSUM	-	4	4_4	0.1	0.1	0.2	0.1	[0, 16)	0.2	0.8	5	0.004	2.5	0.92
	CNN/DM	-	4	4_4	0.3	0.2	0.3	0.1	[0, 16)	0.4	0.8	5	0.002	2.5	0.9
	De⇒En	-	8	4_2	0.1	0.1	0.3	0.1	[0, 16)	0.1	0.8	5	0.004	2.5	0.9
	En⇒De	-	4	4_2	0.1	0.1	0.3	0.1	[0, 16)	0.1	0.8	5	0.004	2.5	0.9
	Zh⇒En	-	4	4_2	0.2	0.1	0.5	0.1	[0, 16)	0.1	0.8	5	0.004	2.5	0.9
	En⇒Zh	-	4	4_4	0.2	0.2	0.1	0.1	[16,32)	0.1	0.8	5	0.004	2.5	0.9
	CQA	-	4	8_4	0.4	0.2	0.2	0.1	[16,32)	0.2	0.85	5	0.004	3.0	0.9
	SQA	-	4	8_2	0.5	0.3	0.3	0.7	[0, 16)	0.3	0.85	5	0.0006	5.0	0.92
	Wikinews	-	4	8_2	0.6	0.5	0.6	0.9	[16,32)	0.8	0.85	10	0.0003	2.5	0.92
	Wikitext	-	4	4_2	0.6	0.4	0.6	0.9	[0, 16)	0.9	0.8	20	0.002	5.0	0.95
	Book	-	4	4_2	0.6	0.4	0.5	0.9	[0, 16)	0.8	0.95	50	0.0006	3.0	0.2
Llama2-7B-Chat	HumanEval	-	8	8_4	0.5	0.4	0.5	0.9	[0, 16)	0.1	0.9	5	0.0003	3.0	0.9
	MBPP	-	8	8_2	0.3	0.2	0.2	0.1	[0, 16)	0.3	0.8	5	0.002	4.0	0.95
	GSM8K	-	8	4_2	0.3	0.4	0.1	0.7	[0, 16)	0.5	0.8	10	0.0009	5.0	0.95
	XSUM	-	8	8_2	0.3	0.4	0.4	0.1	[0, 16)	0.5	0.85	10	0.0009	5.0	0.2
	CNN/DM	-	8	8_2	0.1	0.1	0.1	0.3	[0, 16)	0.5	0.85	5	0.004	2.5	0.92
	CQA	-	4	8_2	0.2	0.5	0.4	0.5	[16,32)	0.5	0.85	50	0.002	4.0	0.92
	SQA	-	4	8_2	0.1	0.4	0.5	0.1	[0, 16)	0.1	0.85	100	0.0009	4.0	0.95
	Wikinews	-	4	8_2	0.4	0.5	0.3	0.9	[16,32)	0.5	0.8	5	0.002	5.0	0.95
	Wikitext	-	4	8_2	0.6	0.6	0.2	0.7	[16,32)	0.1	0.85	20	0.0009	4.0	0.95
	Book	-	8	4_2	0.4	0.4	0.5	0.7	[16,32)	0.8	0.85	10	0.002	3.0	0.95
	FActScore	-	8	4_2	0.1	0.2	0.4	0.9	[16,32)	0.5	0.8	5	0.0006	5.0	0.95
	AlpacaEval	-	8	4_2	0.1	0.2	0.4	0.9	1	0.5	0.8	5	0.0006	5.0	0.95

Table 27: Optimal hyperparameter settings in Table 1.

Model	Dataset	Deterministic Methods								Stochastic Methods					
		Greedy	**BS**	**DBS**	**CS**	**FSD**	**FSD-d**	**CD**	**DoLa**	**Temp**	**Top-p**	**Top-k**	η	**Miro**	**Typical**
7B	MBPP	-	4	8_2	0.3	0.2	0.4	0.3	[16,32)	0.3	0.8	5	0.002	4.0	0.9
	GSM8K	-	8	4_2	0.4	0.3	0.4	0.3	[0,16)	0.2	0.8	5	0.004	5.0	0.9
	Wikinews	-	4	8_2	0.6	0.5	0.6	0.9	[16,32)	0.8	0.85	10	0.0003	2.5	0.92
13B	MBPP	-	4	8_2	0.2	0.3	0.4	0.3	[0,20)	0.3	0.85	5	0.002	2.5	0.2
	GSM8K	-	4	8_2	0.2	0.3	0.4	0.3	[0,20)	0.1	0.8	5	0.002	2.5	0.2
	Wikinews	-	4	4_2	0.5	0.4	0.3	0.9	[0,20)	0.7	0.95	50	0.004	5.0	0.9
70B	MBPP	-	8	4_4	0.6	0.1	0.4	0.3	[0,20)	0.1	0.8	5	0.0003	5.0	0.2
	GSM8K	-	4	4_2	0.2	0.2	0.5	0.9	[0,20)	0.4	0.8	5	0.0006	5.0	0.2
	Wikinews	-	4	8_2	0.6	0.1	0.6	0.9	[60,80)	0.9	0.85	50	0.002	3.0	0.2
7B-chat	MBPP	-	4	8_2	0.3	0.2	0.4	0.3	[16,32)	0.3	0.8	5	0.002	4.0	0.9
	GSM8K	-	8	4_2	0.3	0.4	0.1	0.7	[0,16)	0.5	0.8	10	0.0009	5.0	0.95
	Wikinews	-	4	8_2	0.4	0.5	0.3	0.9	[16,32)	0.5	0.8	5	0.002	5.0	0.95
13B-chat	MBPP	-	8	8_2	0.3	0.3	0.4	0.9	[20,40)	0.3	0.95	5	0.0003	4.0	0.2
	GSM8K	-	8	8_2	0.4	0.2	0.5	0.7	[20,40)	0.4	0.9	50	0.004	5.0	0.92
	Wikinews	-	8	4_4	0.5	0.4	0.6	0.7	[20,40)	0.5	0.8	50	0.0006	3.0	0.9
70B-chat	MBPP	-	8	8_2	0.6	0.2	0.1	0.9	[60,80)	0.6	0.9	5	0.0006	2.5	0.2
	GSM8K	-	4	8_2	0.4	0.2	0.4	0.1	[60,80)	0.3	0.8	20	0.004	2.5	0.9
	Wikinews	-	4	4_2	0.4	0.6	0.2	0.9	[0,20)	0.6	[60,80)	5	0.002	2.5	0.95

Table 28: Optimal hyperparameter settings in Table 4.

Model	Dataset	Deterministic Methods								Stochastic Methods					
		Greedy	**BS**	**DBS**	**CS**	**FSD**	**FSD-d**	**CD**	**DoLa**	**Temp**	**Top-p**	**Top-k**	η	**Miro**	**Typical**
13B-INT4	MBPP	-	4	4_4	0.1	0.2	0.4	0.3	[0,20)	0.3	0.8	10	0.004	4.0	0.92
	GSM8K	-	4	4_2	0.3	0.3	0.3	0.3	[0,20)	0.4	0.8	5	0.0003	2.5	0.2
	Wikinews	-	4	4_2	0.3	0.5	0.5	0.9	[0,20)	0.9	0.8	100	0.0006	3.0	0.95
13B-INT8	MBPP	-	4	8_4	0.5	0.3	0.4	0.3	[0,20)	0.4	0.8	5	0.0006	4.0	0.92
	GSM8K	-	4	4_2	0.6	0.2	0.5	0.3	[0,20)	0.1	0.8	5	0.004	3.0	0.9
	Wikinews	-	4	4_2	0.5	0.2	0.6	0.9	[0,20)	0.9	0.85	10	0.0009	3.0	0.95
13B-Chat-INT4	MBPP	-	8	8_4	0.1	0.3	0.2	0.1	[20,40)	0.1	0.9	5	0.0009	2.5	0.2
	GSM8K	-	8	8_2	0.1	0.3	0.5	0.1	[20,40)	0.5	0.85	10	0.004	3.0	0.92
	Wikinews	-	8	4_2	0.2	0.6	0.6	0.9	[20,40)	0.7	0.95	5	0.0006	2.5	0.2
13B-Chat-INT8	MBPP	-	4	8_2	0.4	0.2	0.5	0.3	[20,40)	0.9	0.9	5	0.0009	3.0	0.92
	GSM8K	-	4	8_2	0.3	0.3	0.3	0.3	[20,40)	0.5	0.8	100	0.0009	3.0	0.95
	Wikinews	-	8	4_4	0.5	0.1	0.4	0.7	[20,40)	0.8	0.95	5	0.002	3.0	0.2

Table 29: Optimal hyperparameter settings in Table 5.

Model	Dataset	Deterministic Methods						Stochastic Methods					
		Greedy	**BS**	**DBS**	**CS**	**FSD**	**FSD-d**	**Temp**	**Top-p**	**Top-k**	η	**Miro**	**Typical**
CodeLlama-7b	MBPP	-	4	4_4	0.3	0.1	0.3	0.6	0.8	5	0.004	4.0	0.2
	GSM8K	-	4	8_2	0.3	0.2	0.5	0.2	0.8	5	0.004	5.0	0.95
	Wikinews	-	8	4_2	0.6	0.6	0.6	0.9	0.9	50	0.004	2.5	0.2
CodeLlama-7b-Instruct	MBPP	-	4	8_4	0.5	0.3	0.2	0.3	0.8	5	0.002	2.5	0.9
	GSM8K	-	4	8_2	0.3	0.1	0.4	0.2	0.8	5	0.0003	5.0	0.2
	Wikinews	-	4	8_2	0.3	0.2	0.3	0.8	1	10	0.0003	2.5	0.95
Qwen-7B	MBPP	-	4	4_2	0.3	0.1	0.1	0.1	0.85	5	0.0009	2.5	0.2
	GSM8K	-	4	8_2	0.5	0.1	0.5	0.1	0.8	5	0.002	2.5	0.2
	Wikinews	-	4	8_4	0.1	0.3	0.3	0.4	0.8	50	0.0009	2.5	0.92
Qwen-7B-Chat	MBPP	-	4	8_4	0.4	0.1	0.1	0.2	0.85	50	0.004	3.0	0.2
	GSM8K	-	8	8_4	0.1	0.3	0.2	0.2	0.85	10	0.0009	3.0	0.2
	Wikinews	-	4	4_4	0.3	0.3	0.1	0.3	0.95	10	0.0006	3.0	0.92
MPT-7b	MBPP	-	8	8_2	0.5	0.1	0.3	0.1	0.85	5	0.004	3.0	0.92
	GSM8K	-	4	4_2	0.5	0.3	0.5	0.2	0.9	5	0.004	5.0	0.9
	Wikinews	-	4	8_2	0.6	0.4	0.5	0.8	0.8	50	0.0006	5.0	0.95
MPT-7b-Instruct	MBPP	-	8	8_2	0.4	0.3	0.4	0.2	0.8	5	0.0009	4.0	0.95
	GSM8K	-	4	4_4	0.2	0.3	0.4	0.5	0.9	10	0.0006	2.5	0.2
	Wikinews	-	4	8_4	0.3	0.3	0.5	0.1	0.85	50	0.002	4.0	0.2
Mistral-7B	MBPP	-	4	4_4	0.1	0.1	0.4	0.9	0.95	20	0.004	2.5	0.95
	GSM8K	-	8	8_2	0.1	0.6	0.2	0.4	0.85	20	0.002	2.5	0.2
	Wikinews	-	4	4_2	0.5	0.5	0.4	0.8	0.9	20	0.0009	5.0	0.95
Mistral-7B-Instruct	MBPP	-	4	8_4	0.4	0.5	0.2	0.4	0.85	100	0.004	2.5	0.95
	GSM8K	-	4	8_2	0.3	0.1	0.6	0.5	0.8	10	0.0009	2.5	0.2
	Wikinews	-	4	8_4	0.5	0.2	0.1	0.7	0.95	5	0.0003	3.0	0.9
deepseek-moe-16b-base	MBPP	-	4	4_2	0.4	0.3	0.1	0.3	0.9	50	0.0003	5.0	0.9
	GSM8K	-	4	4_2	0.5	0.3	0.2	0.4	1	10	0.004	3.0	0.95
	Wikinews	-	4	4_2	0.4	0.2	0.5	0.7	0.9	10	0.0003	3	0.9
deepseek-moe-16b-chat	MBPP	-	4	8_2	0.1	0.4	0.2	0.6	0.8	5	0.002	4.0	0.92
	GSM8K	-	4	8_2	0.2	0.3	0.1	0.4	0.8	5	0.004	5.0	0.9
	Wikinews	-	8	4_4	0.6	0.4	0.5	0.9	0.85	50	0.0006	5	0.95
Llama-3-8B	MBPP	-	4	8_2	0.3	0.2	0.3	0.9	1	20	0.002	3.0	0.2
	GSM8K	-	4	8_2	0.2	0.1	0.5	0.4	1	50	0.0009	4.0	0.92
	Wikinews	-	8	8_2	0.3	0.3	0.5	0.8	0.8	100	0.0003	3	0.2
Llama-3-8B-Instruct	MBPP	-	8	4_4	0.6	0.6	0.3	0.2	0.8	20	0.0006	2.5	0.2
	GSM8K	-	8	4_2	0.1	0.4	0.3	0.1	1	5	0.002	3.0	0.95
	Wikinews	-	8	4_4	0.3	0.1	0.6	0.2	0.8	100	0.0006	5	0.95
vicuna-7b-v1.5	MBPP	-	8	8_2	0.1	0.4	0.4	0.3	0.85	5	0.0006	5.0	0.9
	GSM8K	-	8	8_2	0.2	0.1	0.1	0.4	0.8	5	0.0009	2.5	0.2
	Wikinews	-	8	4_4	0.6	0.6	0.2	0.5	0.9	10	0.0009	3.0	0.95

Table 30: Optimal hyperparameter settings in Table 26.

Model	Setting	Deterministic Methods								Stochastic Methods					
		Greedy	**BS**	**DBS**	**CS**	**FSD**	**FSD-d**	**CD**	**DoLa**	**Temp**	**Top-p**	**Top-k**	η	**Miro**	**Typical**
Llama2-7B	$\text{Score}_{\text{best}}$	78.47	80.97	79.26	94.49	96.71	97.52	92.90	89.94	95.36	80.61	74.70	71.44	67.68	75.68
	Score_{λ}	78.47	80.68	76.53	91.85	94.32	94.76	83.48	86.34	83.77	79.65	71.80	69.61	64.93	74.15
	Param.	-	4	4_2	0.4	0.2	0.4	0.3	0	0.6	0.8	0.5	0.004	2.5	0.9
	Drop	0.00	0.29	2.74	2.64	2.39	2.76	9.42	3.60	11.59	0.96	2.89	1.83	2.76	1.53
Llama2-7B-Chat	$\text{Score}_{\text{best}}$	87.90	96.39	95.79	91.61	94.97	93.50	92.75	74.99	94.97	91.83	91.82	91.45	88.71	92.41
	Score_{λ}	87.90	95.03	95.54	89.70	91.37	91.02	89.40	73.32	91.06	89.57	89.41	88.88	86.29	90.91
	Param.	-	8	8_2	0.2	0.3	0.3	0.7	0	0.4	0.85	0.5	0.002	4	0.95
	Drop	0.00	1.63	0.10	2.09	4.22	2.68	3.76	1.69	4.68	2.41	2.89	2.78	2.36	1.80

Table 31: Hyperparameter Sensitivity. $\text{Score}_{\text{best}}$ and the best Score_{λ} with their optimal hyperparameters on Llama2-7B and Llama2-7B-Chat.

GSM8K

Instruction: Please answer the math questions below.
Luke is buying fabric for new curtains. There are five windows. Each window is 35 inches wide and Luke needs to buy fabric equal to 2 times the total width of the windows. How much fabric should he buy?
You need to first take step-by-step reasoning and then give the final result.

Response: Sure, I'd be happy to help you with that! Here's the step-by-step reasoning:
1. The total width of the five windows is 5 x 35 = 175 inches.
2. Luke needs to buy fabric equal to 2 times the total width of the windows, so he needs to buy fabric equal to 2 x 175 = 350 inches.
Therefore, Luke should buy 350 inches of fabric.
I hope this helps! Let me know if you have any other questions.b-1

XSUM

Instruction: Article: The 32-year-old had been playing in the Isthmian Premier Division with Leatherhead following his release by Newport at the end of last season.
Pidgeley has made 260 appearances in spells with nine clubs, including Chelsea, Watford and Millwall.
Forest Green are currently second in the National League table, one point behind leaders Cheltenham Town.
Pidgeley could make his Rovers debut when they host Aldershot on Friday.

Summarize the above article in 1 sentence.

Response: Former Newport player, James Pidgeley, who has played for 9 different clubs including Chelsea, Watford, and Millwall, has signed with Forest Green Rovers in the National League and could make his debut against Aldershot on Friday. (No need for a question)oundsMS))))))

Wikinews

Instruction: Please help me complete the text continuation based on the following content.

Snooker player Alex Higgins, nicknamed "Hurricane", was found dead in his apartment in Donegall Road, Belfast, Northern Ireland on Saturday. He was

Response: Sure, here is a possible continuation of the text:
Snooker player Alex Higgins, nicknamed "Hurricane", was found dead in his apartment in Donegall Road, Belfast, Northern Ireland on Saturday. He was 61 years old and had been struggling with health problems in recent years. The cause of death is currently unknown, but police are treating it as a suspicious circumstances. Higgins was one of the most iconic figures in snooker history, known for his flamboyant style and fierce competitiveness on the table. He won the World Snooker Championship four times and was inducted into the World Snooker Hall of Fame in 2004. His death has sent shockwaves through the snooker community, with many players and fans paying tribute to his legacy.b
Please let me know if you would like me to make any changes or add anything else. to continue the text. To Complete the Text... To Complete the Text..
I hope this helps! Let me know if you have any other questions or requests.. MS.

Table 32: Failed Examples of DoLa.