How Much Context Does Natural Language Actually Require? An Analysis Using LLMs as Statistical Oracles

Anonymous Authors[1]

Abstract

Despite the growing trend towards large-context transformer models, key questions remain about how much context is truly required for accurate language modeling. We explore this by treating large language models as statistical oracles and measuring the smallest prefix needed to replicate full-context next-token predictions. Using samples from diverse natural text sources, we evaluate minimal context length requirements across various decoding strategies using correctness and support set overlap metrics. Under greedy decoding, we find that over 80% of tokens require less than 10% of the most recent context to yield identical predictions. For general sampling strategies, we define Recall and Risk metrics to assess context dependence, and find that dynamic strategies offer higher support coverage at low percentiles—while also increasing Risk due to broader supports at shorter contexts.

1 Introduction

When prompted to continue a piece of text, how much of the preceding context does a human actually rely on? Do they focus on recent words and local coherence, or plan with a broader, narrative-wide perspective? For example, when writing a story, do they recall events from earlier chapters or rely mainly on recent developments? While this question is difficult to study rigorously in humans, large language models offer a testable analogue. Given a sequence of text, we can ask: how far back must a model look to predict the next token accurately? This question lies at the intersection of interpretability, sampling, and architecture. Although modern transformers can attend to thousands of tokens, it's unclear how much of that capacity is truly used at inference time, and whether predictions depend on distant or local spans. This is especially relevant as language models now rival or surpass human-level performance in many tasks. We present a systematic method for quantifying the context length needed for next-token prediction and analyze how it varies across models, datasets, and decoding strategies. See Appx. 5 for further related-work and motivations.

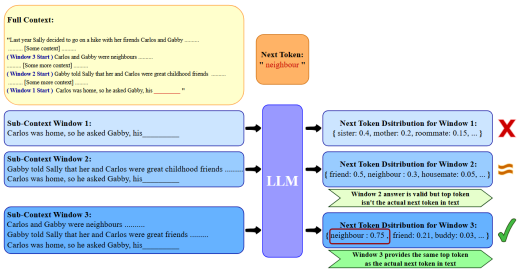

Figure 1: **Minimum Context Length (MCL) Selection:** A scenario illustrating our MCL selection strategy. The example also highlights the need for *distributional awareness*. Although Window-2 yields valid predictions, MCL rejects it and selects Window-3 as the minimal context correctly predicting the actual next token in the dataset. Our distributionally-aware MCL (DaMCL) metric resolves such issues.

1.1 Experimental Setup

We use LLaMA-3-8B (Grattafiori et al., 2024), Mistral-7B (v0.1) (Jiang et al., 2023) and Qwen2-7B (Yang et al., 2024) as oracle language models and focus on natural language documents, including Reddit Writing Prompts (Fan et al., 2018), CNN/DailyMail news articles (Hermann et al., 2015; Nallapati et al., 2016), U.S. Government reports from GovReport (Huang et al., 2021), and Wikipedia articles from WikiText-103 (Merity et al., 2016). For CNN/DailyMail, Writing Prompts, and WikiText-103, we use the first 1000 tokens of each document; for GovReport, we use the first 4000 tokens. See Appx. 6.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

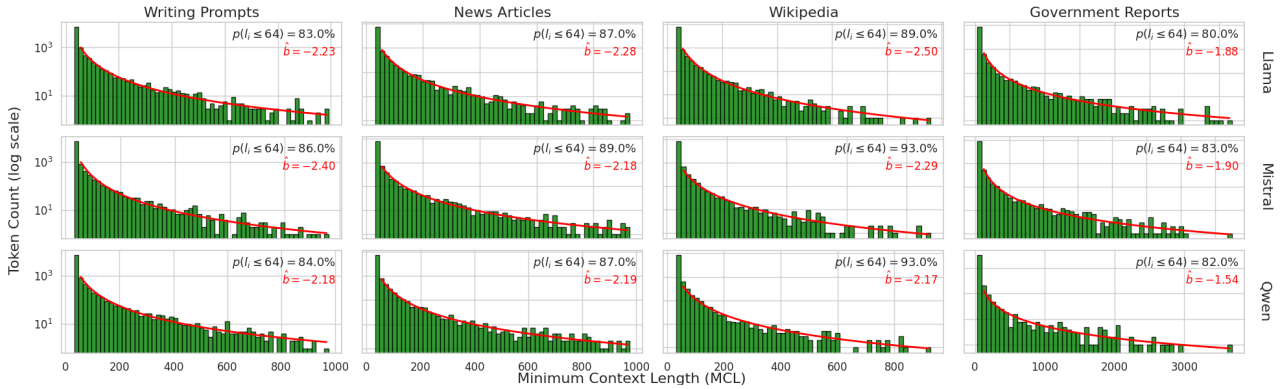

Figure 2: **Distribution of MCL:** Minimum context window needed to confidently predict the next token for sampled contexts across benchmark datasets and LLMs. $\hat{b}$ denotes the slope of the log-log fit. Most predictions resolve with much smaller context windows across models and datasets. Results are consistent across datasets and LLMs.

1.2 Notation

Boldface $\boldsymbol{a}$ differentiates vectors from scalars a. We let $\boldsymbol{a}[i]$ denote the i-th entry of $\boldsymbol{a}$. For integers $i \leq j$, $[i:j]$ denotes the set $\{i, i+1, \ldots, j\}$. For brevity, we write $[:i]$ or simply $[i]$ for $[1:i]$. For $\boldsymbol{a} \in \mathbb{R}^n$, we denote the suffix $\boldsymbol{a}[(n-l):n]$ of length $l+1$ simply as $\boldsymbol{a}[-l:]$. For $p \in (0,1)$, we denote $\boldsymbol{a}[\%p:]$ the suffix consisting of the last p fraction of entries, specifically $\boldsymbol{a}[\lceil(1-p)n\rceil : n]$. We represent sequences of tokens as vectors and the above notations apply. We let $|\boldsymbol{a}|$ denote the vector/sequence's length, i.e., the number of its entries. Δ^n denotes the probability simplex in $\mathbb{R}^n$.

We consider datasets consisting of stories, articles, etc., which we generically refer to as documents. Documents are tokenized with respect to vocabulary $\mathcal{V}$ of $V \triangleq |\mathcal{V}|$ tokens. We let $\mathbf{s} = [t_1, t_2, \ldots, t_n] \in \mathcal{V}^n$ be a document of n tokens, representing a complete story, news article, or government report from the dataset. We let $\mathbf{s}_{[i]} = [t_1, t_2, \ldots, t_i]$ denote the context window of the first i tokens.

For $k \leq n$, we define operator $\mathsf{Top}_k(\cdot) : \mathbb{R}^n \to \binom{[n]}{k}$, returning the indices of the top k entries of its argument, where $\binom{[n]}{k}$ is the set of all k-element subsets of $[n]$. Let $\mathsf{Top}_k^{(l)}(\cdot) \in [n]$ denote the l-th largest element in this set when sorted in decreasing order, with $l \in [k]$. Note that $\mathsf{Top}_n^{(l)}(\boldsymbol{a})$ is the index of the l-th largest entry of $\boldsymbol{a}$ when sorted in decreasing order.

Let $\pi_{\boldsymbol{\theta}} : \mathcal{V}^* \to \Delta^V$ denote an auto-regressive language model with parameters $\boldsymbol{\theta}$. Given input sequence $\mathbf{s}$, the model outputs a probability distribution over $\mathcal{V}$: $\pi_{\boldsymbol{\theta}}(\mathbf{s}) \in \Delta^V$. Moreover, for $k \leq V$, denote the composition of the Top-k operator with the model output given an input sequence as $\mathsf{Top}_{k,\boldsymbol{\theta}}(\cdot) : \mathcal{V}^* \to \binom{[V]}{k}$. That is, the operator selects the k most likely next tokens according to the model's output distribution. Similarly, define the confidence with which the model predicts the next token as the probability gap between the top-ranked and second-ranked tokens. Both definitions are summarized by the expressions: $\mathsf{Top}_{k,\boldsymbol{\theta}}(\mathbf{s}) := \mathsf{Top}_{k,\boldsymbol{\theta}}(\pi_{\boldsymbol{\theta}}(\mathbf{s}))$ and $\triangle\mathsf{Conf}_{\boldsymbol{\theta}}(\mathbf{s}) := \pi_{\boldsymbol{\theta}}[\mathsf{Top}_{2,\boldsymbol{\theta}}^{(1)}(\mathbf{s})] - \pi_{\boldsymbol{\theta}}[\mathsf{Top}_{2,\boldsymbol{\theta}}^{(2)}(\mathbf{s})] \geq 0$.

2 Least Context for Prediction

Consider the following question: *For a given randomly sampled context and next-token, what is the minimum sub-context needed to predict the actual next token correctly?*

2.1 Minimal Context Length

As a first step, we focus on sequences where the oracle LLM **correctly** and **confidently** predicts the next token from text using greedy decoding. We set a confidence threshold $\delta \in [0,1]$, meaning the top token has at least δ higher probability than the second-best. Using this, we define:

Definition 2.1. The **Minimal Context Length** (MCL) of sequence $\mathbf{s}$ given the true next token t is defined as the length of the smallest prefix of the sequence such that the model output given the prefix is correct and confident (with parameter $\delta \in [0,1]$). Formally,

$$\mathsf{MCL}\,(\mathbf{s}|t) := \arg\min_{l \in |\mathbf{s}|} \left\{ l \;\middle|\; \mathsf{Top}_{1,\boldsymbol{\theta}}(\mathbf{s}_{[-l:]}) = t, \;\; \triangle\mathsf{Conf}_{\boldsymbol{\theta}}(\mathbf{s}_{[-l:]}) \geq \delta \right\}$$

In essence, $\mathsf{MCL}\,(\mathbf{s}|t)$ represents the minimum number of tokens the model needs to consider from the end of sequence $\mathbf{s}$ to

confidently and correctly predict the next token t.

2.2 Experimental Details

We form sequences (contexts) by parsing documents $\mathbf{d}$ from the datasets. For the *Writing Prompts*, *News Articles* and *Wikipedia* datasets, we sample 100 unique documents and set maximum document length $n = 1000$ by truncating documents to their first 1000 tokens. For *Government Reports*, we set $n = 4000$ tokens as these documents typically rely more on long-context information. From each document, we sample 100 contexts $\mathbf{s}_{[i]}$ of varying lengths with $i \in [32, n-1]$ and their respective next token t_{i+1}. When sampling values of i, we ensure uniform distribution across document positions, avoiding bias toward either shorter or longer context windows. This approach yields 10,000 unique contexts (100 contexts × 100 documents) and their respective next tokens for each dataset. Our sampling criterion requires that the model correctly and confidently predicts the next token when given the full context, i.e., $\mathsf{Top}_{1,\boldsymbol{\theta}}(\mathbf{s}_{[i]}) = t_{i+1}$ and $\triangle\mathsf{Conf}_{\boldsymbol{\theta}}(\mathbf{s}_{[i]}) \geq \delta$.

Note that for these selected sequences, MCL represents the shortest suffix of the original context for which the model output remains both correct and confident. **A higher value of MCL implies that the model requires information from earlier in the context to predict the next token correctly, while a smaller value indicates greater reliance on local information from the most recent tokens.** For concreteness, in our experiments we choose $\delta = 0.2$.

To determine the MCL, we evaluate a model's predictions using increasing context window sizes $l \in \{32, 48, 64, \ldots, |\mathbf{s}|\}$, starting from 32 tokens and incrementing by 16. This choice reflects prior work suggesting 32 tokens capture local context beyond n-gram statistics (Liu et al., 2025; Fang et al., 2025). For each window size, we examine the model's next-token distribution and stop once it confidently predicts the correct next token or the full context is reached. In practice, we provide the full input to preserve positional encoding and simulate truncated contexts via attention masking. While using only the truncated input yields similar behavior, it disrupts positional encoding and may conflate prediction differences with positional shifts rather than true contextual effects—potentially confounding interpretation of minimal context requirements.

2.3 Results and Discussion

As shown in Fig. 2, the distribution of $\mathsf{MCL}\left(\mathbf{s}_{[i]}|t_{i+1}\right)$ is highly skewed (with the histogram y-axis in log scale), indicating that the model requires only the last 32–64 tokens for the majority of contexts ($\geq$ 80–90%) to confidently predict the next token ($\mathsf{MCL}\left(\mathbf{s}_i|t_{i+1}\right) \leq 64$). To quantify this behavior, we examine the slope $\hat{b}$ of the best-fitting line in log-log space (i.e., for $y = a \cdot x^b$). We exclude the first dominating bin MCL ≤ 32 tokens to better capture the trend. We find that $\hat{b}$ typically hovers around 2, suggesting that LLMs rely primarily on recent local tokens for prediction. Notably, this trend persists even for Government Reports—commonly used as a long-context benchmark (Bai et al., 2024)—albeit with a shallower slope. This pattern is consistent across all three models, further supporting the generality of the observation. . These findings align with the motivations in (Fang et al., 2025), reinforcing the idea that local information is often sufficient for confident next-token prediction from the model's perspective. Finally, it is worth pointing that the MCL distribution over datasets appears consistent across various LLMs.

3 Distributional Awareness

MCL evaluates whether a model can predict the actual next token from the dataset, assuming a single ground-truth continuation. However, natural language often permits multiple valid next tokens, and models may assign high probability to plausible alternatives not in the dataset. Moreover, greedy decoding fails to reflect how generation methods work—many rely on sampling strategies that consider sets of probable tokens rather than the top token. These limitations motivate a broader MCL formulation that (1) relies on the model's next-token distribution rather than a single ground-truth, and (2) incorporates the dynamics of different sampling strategies. We elaborate on these distribution-aware requirements in Appx 7.

3.1 Distribution Aware MCL

Consider a decoding method ϕ that takes as input $\boldsymbol{\pi}(\mathbf{s})$ for some input sequence $\mathbf{s}$, truncates the distribution, and outputs a set of valid tokens which we denote $\mathcal{A}_{\mathbf{s},\phi}$. We will refer to this $\mathcal{A}_{\mathbf{s},\phi}$ as the next-token **Support Set** for the context $\mathbf{s}$ when sampling with ϕ. Furthermore, we define the recall metric from set A to set B as:

$$\mathsf{Recall}\left(A \mid B\right) := |A \cap B| \big/ |B| \in [0, 1]\,,$$

measuring the proportion of elements from set B that are contained in set A. Recall $= 1$, indicates that the entirety of elements in B are included in A, $B \subseteq A$. When considering the support set of distributions, the higher this value, the more set A covers elements in set B. We use Recall to define a notion of MCL that does not depend on the specific next-token of

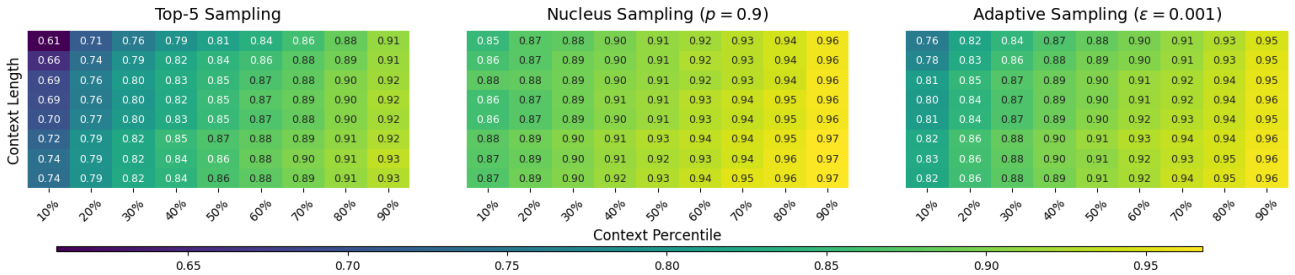

Figure 3: **Recall Trend:** Average Recall score for different context lengths $i \in [200, 1000]$ (on the y-axis) and context percentiles $p \in [10\% \ldots, 90\%]$ (on the x-axis) for Writing Prompts samples using Mistral-7B as the oracle.

a given sequence from the dataset but rather focuses on the valid support set as per the sampling strategy.

Definition 3.1. The **Distribution-aware Minimal Context Length** (DaMCL) of a sequence $\mathbf{s}$, as measured by a statistical oracle LLM with decoding strategy ϕ, is defined as the length of the smallest prefix such that the decoding method's truncation set (aka the support set) $\mathcal{A}_{\mathbf{s}_{[-l:]},\phi}$ for the prefix covers that of the full context $\mathcal{A}_{\mathbf{s},\phi}$. Formally, $\mathsf{DaMCL}_{\phi}(\mathbf{s})$ is defined to be the

$$\arg\min_{l \in |\mathbf{s}|} \left\{ l \;\middle|\; \mathsf{Recall}\left(\mathcal{A}_{\mathbf{s}_{[-l:]},\phi} \mid \mathcal{A}_{\mathbf{s},\phi}\right) = 1 \right\} .$$

This value represents the minimal context length required for the short context's support set to fully contain the long context's next token support. Note that for greedy decoding, when restricted to sequences $\mathbf{s}$ where the top-1 token of the oracle output matches the true next token t in the dataset, $\mathsf{DaMCL}_{K=1}(\mathbf{s})$ reduces to $\mathsf{MCL}(\mathbf{s}|t)$. Thus, this definition is more general while satisfying the two desiderata from the previous section.

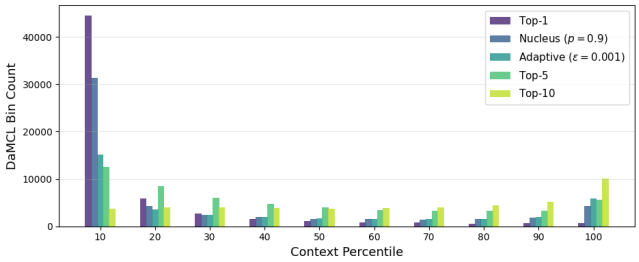

Figure 4: **Impact of Sampling:** Distribution of $\mathsf{DaMCL}_K(s_i)$ for $K \in \{1, 5, 9\}$, along with Nucleus and Adaptive sampling. While DaMCL for $K = 1$ and Nucleus behave similarly to MCL (greedy decoding), increasing K shifts the distribution toward requiring longer context spans for full Recall.

3.2 Experimental Details

We evaluate several decoding strategies: Top-K sampling (K=1 for greedy) with $K \in \{1, \ldots, 9\}$ (Radford et al., 2019; Fan et al., 2018), nucleus sampling with p=0.9 (Holtzman et al., 2020), and adaptive sampling with ϵ=0.001 (Zhu et al., 2024). To ensure comparability across context sizes, we use truncated windows based on percentiles of the full context (e.g., last 10% to 100%) rather than fixed lengths. Positions i are limited to [400, 4000] for government reports and [200, 1000] otherwise. This percentile-based setup allows fairer comparisons and lowers experimental overhead. For Fig. 4, we combine results across both models and all three datasets (*Writing Prompts*, *News Articles*, *Government Reports*), as distributions are consistent across settings.

3.3 Results and Discussion

Looking at static sampling methods, namely Top-k, we observe that increasing the support set size skews the DaMCL values, requiring the model to use larger portions of the context to achieve full Recall (Fig. 4). For most contexts, when using $k = 1$ (greedy sampling) or nucleus sampling, local sub-contexts ($l = 32$) are often sufficient for full Recall. In contrast, adaptive sampling typically falls between Top-1 and Top-5, and exhibits a more U-shaped distribution of DaMCL values—suggesting that accurate predictions often rely more equally on local or full-context information. These patterns highlight how sensitive context utilization metrics are to the choice of decoding strategy.

Additionally, by analyzing Recall scores across percentiles rather than relying solely on the binary DaMCL threshold, we gain a more nuanced view of how sampling interacts with context length. Fig. 3 shows Recall heatmaps for different sampling methods. Dynamic methods (nucleus and adaptive) yield higher Recall scores—especially at lower percentiles (e.g., 10% or 20%)—compared to the static Top-5 method, indicating that shorter subcontexts more often recover the full-context support. Unlike Top-5, Nucleus and Adaptive sampling also show minimal variation across percentiles within each row, suggesting a robustness of DaMCL to context length under dynamic decoding.

For a final discussion of our work and future directions, please see Appx. 4.

References

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., Dong, Y., Tang, J., and Li, J. Longbench: A bilingual, multitask benchmark for long context understanding, 2024. URL https://arxiv.org/abs/2308.14508.

Basu, S., Ramachandran, G. S., Keskar, N. S., and Varshney, L. R. Mirostat: A neural text decoding algorithm that directly controls perplexity, 2021. URL https://arxiv.org/abs/2007.14966.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., and Katie Millican, e. a. Improving language models by retrieving from trillions of tokens, 2022. URL https://arxiv.org/abs/2112.04426.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation, 2023. URL https://arxiv.org/abs/2306.15595.

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

Fang, L., Wang, Y., Liu, Z., Zhang, C., Jegelka, S., Gao, J., Ding, B., and Wang, Y. What is wrong with perplexity for long-context language modeling?, 2025. URL https://arxiv.org/abs/2410.23771.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., and Angela Fan, e. a. . The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. *NeurIPS*, 2015.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1904.09751.

Huang, L., Cao, S., Wang, L., and Ji, H. Efficient attentional models for document summarization. *arXiv preprint arXiv:2104.07241*, 2021.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Atlas: Few-shot learning with retrieval augmented language models, 2022. URL https://arxiv.org/abs/2208.03299.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. The narrativeqa reading comprehension challenge. *TACL*, 2018.

Liu, J., Min, S., Zettlemoyer, L., Choi, Y., and Hajishirzi, H. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens, 2025. URL https://arxiv.org/abs/2401.17377.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016.

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., and Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Pang, R. Y., Parrish, A., Joshi, N., Nangia, N., Phang, J., Chen, A., Padmakumar, V., Ma, J., Thompson, J., He, H., and Bowman, S. R. Quality: Question answering with long input texts, yes!, 2022. URL https://arxiv.org/abs/2112.08608.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *CoRR*, abs/2108.12409, 2021. URL https://arxiv.org/abs/2108.12409.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864, 2021. URL https://arxiv.org/abs/2104.09864.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Sessa, P. G., Chowdhery, A., Roberts, A., Barua, A., and Alex Botev, e. a. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, W., Dong, L., Cheng, H., Liu, X., Yan, X., Gao, J., and Wei, F. Augmenting language models with long-term memory, 2023. URL https://arxiv.org/abs/2306.07174.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., and Jialong Tang, e. a. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Zhou, Y., Keuper, M., and Fritz, M. Balancing diversity and risk in llm sampling: How to select your method and parameter for open-ended text generation, 2025. URL https://arxiv.org/abs/2408.13586.

Zhu, W., Hao, H., He, Z., Ai, Y., and Wang, R. Improving open-ended text generation via adaptive decoding, 2024. URL https://arxiv.org/abs/2402.18223.

4 Conclusion and Future Work

In this work, we introduced the concept of Minimal Context Length (MCL) to quantify how much prior context a language model needs to confidently predict the next token. Our results show that models often rely on recent local context to replicate full-context predictions. When treating LLMs as statistical oracles for the true language distribution, we highlighted limitations of using the actual next token as a target and instead advocated evaluating against the model's own predictive distribution. We emphasized how decoding strategy and sampling pool size—both part of the oracle's output—can affect minimal context requirements.

These findings open avenues for future work on alternative metrics for contextual understanding. To capture aspects beyond Recall, we introduce the Risk metric in Appx. 8, which quantifies over-validation from subcontexts. This offers insight into how LLMs use context and how this interacts with decoding strategies—impacting interpretability and model design. While our experiments focus on natural language, the methodology extends to domains such as math, code, or biomedical text, potentially revealing domain-specific patterns in context use. Complementary to viewing LLMs as statistical oracles, our study also highlights strengths and limitations of decoding methodology.

5 Discussion and Related Work

Transformer-based language models, first introduced by Vaswani et al. (2017), have become the de facto standard for training large-scale language models, due to the self-attention mechanism's ability to flexibly aggregate information over wide context windows. Recent open-access models such as LLaMA 3 (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), Qwen2 (Yang et al., 2024), and Gemma (Team et al., 2024) support context lengths from 8K to 128K tokens. For reference, these lengths can accommodate entire medium-sized novels within a single context window. Additionally, a wide range of architectural and algorithmic innovations have been proposed to improve long-context modeling, including rotary position encodings (RoPE) (Su et al., 2021), attention linear biases (ALiBi) (Press et al., 2021), and position interpolation (Chen et al., 2023), which enable extrapolation to longer sequences without retraining the model from scratch. Beyond positional encodings, recent approaches such as retrieval-augmented transformers (Borgeaud et al., 2022; Izacard et al., 2022; Wang et al., 2023) aim to improve long-context reasoning by retrieving or caching relevant information from earlier context segments, offering alternatives to simply extending attention length.

Much of the evaluation of a model's contextual understanding has focused on tasks such as question answering, retrieval, and needle-in-the-haystack probing, evaluated on datasets such as NarrativeQA (Kočiský et al., 2018), TriviaQA (Joshi et al., 2017), QuALITY (Pang et al., 2022), and LongBench (Bai et al., 2024). While these benchmarks test a model's ability to extract specific information from distant context, they differ from standard language modeling and tend to be highly task-specific. A recent study by Fang et al. (2025) proposes a method for identifying tokens with long-context dependencies and encourages training-time metrics that distinguish such tokens. While their work focuses on a binary classification of long- vs. short-context tokens, we adopt a more fine-grained perspective: treating the language model as a probabilistic oracle and estimating the minimal context required for each next-token prediction in natural text.

Given our assumption that language models serve as strong proxies for language understanding, it is important to account for the decoding strategy used during inference. A growing body of research has shown that different sampling methods—such as greedy decoding, top-k sampling (Radford et al., 2019; Fan et al., 2018), nucleus (p) sampling (Holtzman et al., 2020), and adaptive techniques (Basu et al., 2021; Zhu et al., 2024)—can substantially influence output diversity, factuality, and calibration. Greedy decoding in particular has been shown to produce degenerate or overly deterministic outputs, while adaptive and dynamic approaches aim to adjust sampling entropy and generate a high-quality, contextually valid subset of tokens (Holtzman et al., 2020; Zhu et al., 2024; Basu et al., 2021). When treating the language model as a statistical oracle for analyzing context usage, it is essential to consider how decoding strategy influences conclusions about effective context length. This perspective may help improve the practical utility of methods such as Fang et al. (2025), which focus primarily on a single sample from the next-token distribution to classify tokens by their context length requirements. Accordingly, we provide a dedicated analysis of how decoding strategies impact context dependence in next-token prediction.

6 Experimental Detail

In this study, we investigate the behavior of pretrained LLMs on natural language datasets composed of human-written narratives and documents. We assume that the models under consideration are sufficiently capable to exhibit reliable performance on next-token prediction and question answering tasks. Specifically, we evaluate two open-weight models:

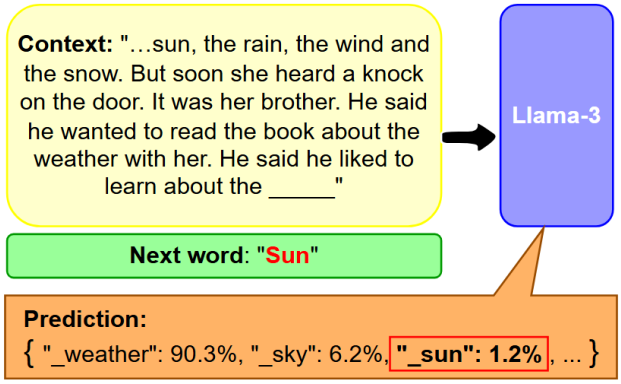

Figure 5: An example illustrating the potential issues when we rely on next token prediction and probability distribution to determine contextual understanding.

Llama-3-8B (Grattafiori et al., 2024) and Mistral-7B (v0.1) (Jiang et al., 2023). Both models share a vocabulary size of approximately $|\mathcal{V}| \approx 32{,}000$, with Llama supporting a maximum context length of 8000 tokens, and Mistral supporting up to 32,768 tokens. All experiments are performed on a V100 Nvidia GPU with 32GB of memory.

Our primary goal is to analyze the minimum context length required for accurate or confident prediction of each token. To this end, we use datasets consisting of plain English text, including narrative and expository writing. For next-token prediction tasks, we use Reddit collected writing prompts (Fan et al., 2018), CNN/DailyMail news articles (Hermann et al., 2015; Nallapati et al., 2016), Wikipedia Articles (Merity et al., 2016), and U.S. Government reports curated from the GovReport dataset [(Huang et al., 2021)]. For CNN/DailyMail, the Writing Prompts and Wikipedia, we slice the first 1000 tokens of any sampled document, and set the cutoff to be the first 4000 tokens for GovReport. These datasets are deliberately chosen for their linguistic simplicity and general domain coverage, avoiding specialized formats such as mathematics or programming code, which may exhibit fundamentally different context dependencies. We leave such extensions to future work.

7 Motivations on DaMCL

In Sec. 2, we posed the question of determining the minimum subcontext prefix needed to predict the next token in a given dataset. A key limitation of this formulation is that it is constrained by the specific realization of the natural language distribution underlying that dataset.

Put simply, given a context, there are often multiple valid next tokens—valid in terms of the underlying (but unknown) distribution of natural language. While we cannot access this true distribution, we have treated pretrained LLMs as statistical oracles. However, in defining MCL in Definition 2.1, we constrain these oracles by evaluating them against only the actual next token from the dataset. Furthermore, we rely solely on greedy decoding, which outputs a single token, thereby underutilizing the model's full predictive distribution as a language oracle.

We summarize the issues as follows:

1. Even if the oracle's top-1 prediction does not match the next token in the source text, i.e., $\mathsf{Top}_{1,\boldsymbol{\theta}}(\mathbf{s}_{[-l:]}) \neq t_{i+1}$, this does not invalidate the model's output or imply a lack of contextual understanding. As shown in Fig. 5, the model assigns high probability to several plausible continuations, even if the dataset token is not ranked first. This suggests that relying solely on the dataset token may mislead any context-length detection method.

2. Using the Top-1 token from the sampling distribution is not always a reliable way to evaluate next-token prediction, as greedy decoding often results in low-quality or repetitive outputs (Holtzman et al., 2020). More recent sampling strategies instead aim to identify a set of valid next tokens (Zhu et al., 2024; Zhou et al., 2025), shifting the focus away from single-token probabilities toward broader support coverage.

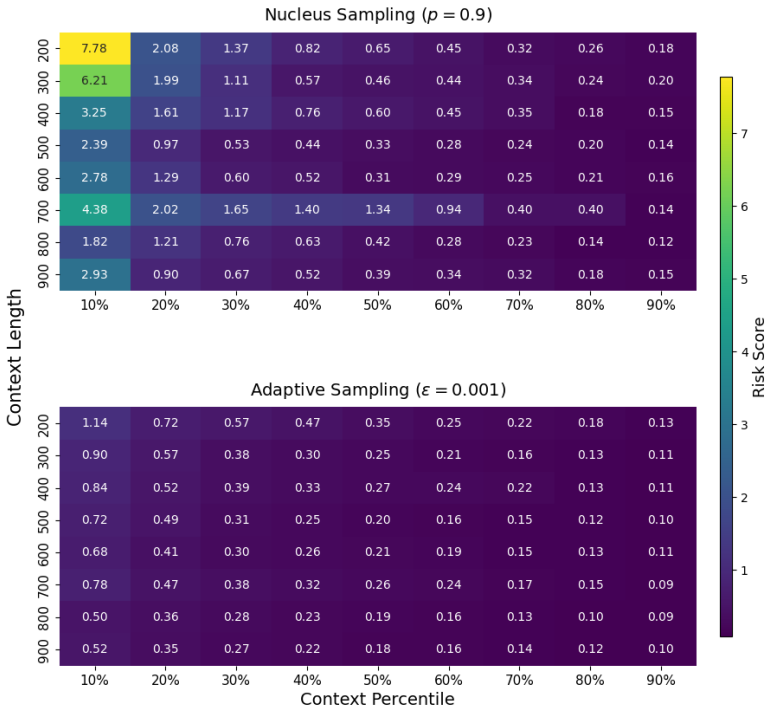

Figure 6: Histogram for Risk distribution across different sampling strategies using Mistral-7B and Reddit Writing Prompts. We can see here that higher risk is associated with lower context percentiles pointing towards possible lack of contextual understanding.

These issues motivate the need for a broader definition of MCL—one that 1) **relies on the model's own next-token distribution rather than the actual next token**, and 2) **accounts for the sampling strategy used during inference**. The goal of DaMCL is to mitigate these limitations and offer a more faithful metric for contextual understanding.

8 Risk, The Missing Metric

While our Recall-based observations are informative, they do not capture an inherent distinction between static methods like Top-k and their dynamic counterparts. In Top-k sampling, the support set is explicitly constrained, potentially excluding some valid tokens, but also reducing noise in next-token prediction. In contrast, dynamic methods—without fixed support size limits—may assign nonzero probability to a much larger set of tokens. This raises a concern: under short-context conditions, could dynamic sampling methods overgenerate, labeling too many tokens as valid due to distributional uncertainty and lack of constraints on the support size ?

In order to have a measure of the amount of samples generated outside the support set for the full-context inference, we keep track of the following metric:

$$\mathsf{Risk}\,(A \mid B) := \frac{|A| - |A \cap B|}{|B|} \in [0, \infty)\,.$$

Effectively, a Risk value of zero implies that $A \subseteq B$. Unlike Recall, Risk is unbounded above; a high Risk value indicates that many elements in A are not found in B. Notably, $\mathsf{Risk}\,(A \mid B) = 0$ and $\mathsf{Recall}\,(A \mid B) = 1$ together imply set equality ($A = B$). However, the two metrics are decoupled: one may observe $\mathsf{Recall}\,(A \mid B) = 1$ (full coverage) while still having $\mathsf{Risk}\,(A \mid B) \gg 1$, meaning A includes many additional, potentially spurious tokens. Conversely, a low Risk with low Recall could simply reflect that A is too small to adequately cover B.

Analogous to our computation of Recall between the subcontext and full-context support sets, we define the Risk metric as $\mathsf{Risk}\left(\mathcal{A}_{\mathbf{s}_{[l:]},\phi} \mid \mathcal{A}_{\mathbf{s},\phi}\right)$. A low Risk value indicates that the subcontext's predicted support closely aligns with the full context's, suggesting that few extraneous tokens are introduced. In contrast, a high Risk value signals that subcontext-based sampling may be overly permissive, validating many tokens that would not appear as plausible under the full context.

As shown in Fig. 6, shorter subcontexts—particularly those in the 10% and 20% percentile ranges—exhibit significantly higher Risk, highlighting the importance of token rejection as context length increases. This may arise from the model becoming more confident in its top predictions with more context, resulting in smaller support sets, or from improved contextual grounding that eliminates tokens which appear valid under limited context.

Interestingly, Top-k sampling provides a natural upper bound on Risk by capping the number of tokens considered valid. This can prevent excessive over-validation by subcontexts. Moreover, while adaptive sampling does exhibit elevated Risk at shorter context lengths, it performs more favorably than nucleus sampling in terms of Risk overall. These findings suggest that Risk, alongside Recall, offers valuable insights into how sampling methods interact with context length. Future work may consider hybrid metrics, such as combining Risk with next-token probability or Recall, to better quantify contextual understanding.