Tagging-Augmented Generation: Assisting Language Models in Finding Intricate Knowledge In Long Contexts

Anwesan Pal[1], Karen Hovsepian[1], Tinghao Guo[2], Mengnan Zhao[2], Somendra Tripathi[3], Nikos Kanakaris[1], George Mihaila[2], Sumit Nigam [4],
[1]AWS AI Labs, [2]Amazon Web Services, [3]Amazon OTS, [4]Amazon Catalog AI,
{anwesanp, khhovsep, tinghg, mengnanz, somendt, nikosk, georgemh, sumnig}@amazon.com
Authors contributed equally

Abstract

Recent investigations into effective context lengths of modern flagship large language models (LLMs) have revealed major limitations in effective question answering (QA) and reasoning over long and complex contexts for even the largest and most impressive cadre of models. While approaches like retrieval-augmented generation (RAG) and chunk-based re-ranking attempt to mitigate this issue, they are sensitive to chunking, embedding and retrieval strategies and models, and furthermore, rely on extensive pre-processing, knowledge acquisition and indexing steps. In this paper, we propose Tagging-Augmented Generation (TAG), a lightweight data augmentation strategy that boosts LLM performance in long-context scenarios, without degrading and altering the integrity and composition of retrieved documents. We validate our hypothesis by augmenting two challenging and directly relevant question-answering benchmarks – NoLiMa and NovelQA – and show that tagging the context or even just adding tag definitions into QA prompts leads to consistent relative performance gains over the baseline – up to 17% for 32K token contexts[1], and 2.9% in complex reasoning question-answering for multi-hop queries requiring knowledge across a wide span of text. Additional details are available at https://sites.google.com/view/tag-emnlp.

1 Introduction

Latest advances in large language models (LLMs) have dramatically expanded their context windows beyond hundreds of thousands of tokens (Chen et al., 2023; Ding et al., 2024; Jin et al., 2024), enabling multi-document and long-context retrieval capabilities such as summarization and question answering. This is handy for technical document understanding, dialogue summarization, and other use cases that demand fine-grained reasoning over increasingly long and complex documents (Liu et al., 2024; Li et al., 2024).

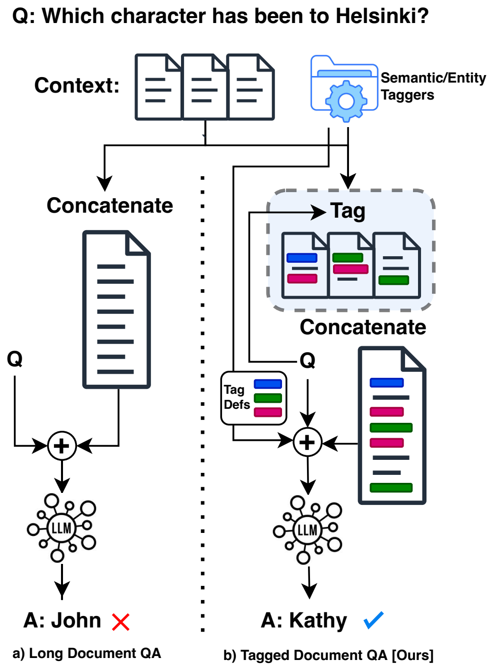


Figure 1: High-level diagram comparing key steps in **(a)** traditional long-document QA and **(b)** the proposed tagging-augmented QA. The diagram highlights the plug-and-play use of imported, domain-specific taggers in an agentic flow. Our framework involves semantically enriching context by means of tagging. Furthermore, we also augment the prompt with an imported list of semantic and entity tag names and definitions to guide the LLM towards recognizing such semantic entities.

Despite these advancements, recent studies on Needle-in-a-Haystack (NIAH) tasks (Kamradt, 2023; Modarressi et al., 2025) reveal that many state-of-the-art LLMs struggle to retrieve and reason over granular details embedded deep within long context, often well below the models' claimed maximum context length. Performance degradation becomes especially evident when the relevant

[1] As of the time of this publication, our best performing model ranks #2 in the NoLiMa leaderboard for 32K context.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 2209–2220
November 4-9, 2025 ©2025 Association for Computational Linguistics

information shares superficial similarity with the query, exposing major challenges for tasks requiring associative and semantic reasoning over extended inputs. Complex narrative benchmarks like NovelQA (Wang et al., 2025) further highlight this limitation as models must reason about interconnected story elements, character arcs and thematic connections distributed across full-length novels.

To address these limitations, we propose Tagging-Augmented Generation (TAG), a lightweight prompting and input augmentation technique that explicitly highlights salient elements in the input context. By injecting structured, LLM-generated semantic annotations (e.g., named entities, topics, discourse roles) directly into the document, our method helps models focus their attention during inference without requiring any architectural modifications or retrieval infrastructure. Compared to traditional retrieval-augmented generation (RAG) pipelines, semantic tagging provides an interpretable, low-latency and infrastructure-free approach to improve LLM performance on long-context reasoning tasks. Furthermore, tags can be cached, so that context that was previously tagged can be retagged without incurring additional compute overhead. Figure 1 shows a high-level comparison between traditional long-context QA and our proposed TAG approach.

Motivation - In real-world scenarios, such as technical support and document analysis, LLMs are often required to process previously unseen, unstructured, and extensive documents without the benefit of pre-indexed retrieval systems. As context lengths grow and becomes more complex, traditional attention mechanisms struggle to maintain focus across distant semantic spans, especially when lexical overlap is minimal (Muennighoff, 2022; Modarressi et al., 2025). Our semantic tagging approach addresses this gap by embedding explicit structural cues within the input, thereby guiding model attention, preserving information retention and enhancing reasoning quality in long and complex context settings.

Contributions - (i) Semantic Tagging and Context Enhancement Framework: We introduce TAG, a flexible framework that employs different tagging mechanisms to produce semantic tags while being agnostic to any named entity recognition (NER) method. TAG supports both traditional and agentic approaches to generate tags without model retraining. To the best of our knowledge, this is the first framework that guides LLMs to perform better in long-context and complex reasoning tasks by semantically enriching the input. **(ii) Benchmark Extension**: We develop augmented versions of two popular long-context benchmarks, NoLiMa (Modarressi et al., 2025) and NovelQA (Wang et al., 2025) to obtain NoLiMa+ and NovelQA+, respectively. These datasets are purpose-built for evaluating LLMs' ability to utilize structural cues in long contexts. **(iii) Empirical Validation**: We present a novel study of empirical results in two different environments – varying context length and complexity settings, using the NoLiMa+ and NovelQA+ benchmarks. Through extensive experiments using two Anthropic Claude models, we demonstrate that semantic tagging improves question-answering accuracy by over 17% for 32K token contexts, and 2.9% in complex reasoning question-answering for multi-hop queries requiring knowledge across a wide span of text, as compared to baselines without tagging.

2 Background

Needle-in-a-Haystack Evaluation - The Needle-in-a-Haystack (NIAH) (Kamradt, 2023) paradigm evaluates long-context understanding by embedding specific information within extensive irrelevant content. Traditional NIAH benchmarks rely heavily on lexical overlap between queries and relevant passages, which may not reflect true semantic reasoning capabilities. NoLiMa (Modarressi et al., 2025) addresses this limitation by introducing a benchmark with minimal lexical cues, requiring latent semantic reasoning across long contexts. The benchmark places 'needles' containing key information within 'haystacks' of random text patches, where questions probe the subject's identity without sharing keywords with the needle. This forces models to rely on semantic associations rather than pattern matching, exposing fundamental limitations in current attention mechanisms even for models with extended context windows.

Complex Narrative Comprehension - While NIAH benchmarks test information retrieval, real-world applications often require understanding complex, interconnected narratives with multiple characters, plotlines and thematic elements distributed across extensive texts. NovelQA (Wang et al., 2025) showcases this through a comprehensive benchmark of 2,305 manually annotated questions spanning 89 full-length novels. This tests models' ability to maintain coherent understand-

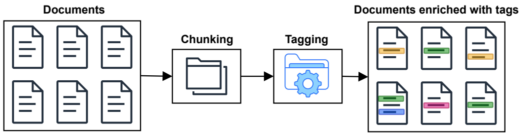


Figure 2: Overview of the proposed TAG framework. Input text is first segmented into de-duplicated chunks using configurable strategies (e.g., sentence-level, paragraph-level, or semantic chunking). These chunks are processed through tagging modules, producing XML-style semantic annotations that preserve document structure while providing explicit attention guidance for downstream tasks. Finally, we have the enriched documents with tags embedded within the text.

ing across extended narratives, requiring multi-hop reasoning about character relationships, plot development and thematic connections. Questions are systematically categorized by complexity (single-hop, multi-hop, detail), enabling granular assessment of long-context comprehension capabilities.

Relevance to our approach - Our work builds on the above insights by introducing semantic tagging, a prompting strategy that explicitly highlights salient information, helping models better navigate intricate contexts without architectural changes.

3 Proposed Approach

TAG Framework - We propose Tagging-Augmented Generation (TAG), a simple and lightweight prompting framework that mitigates long-context and complex-context performance degradation in LLMs through explicit semantic mark-up. TAG addresses the fundamental challenge of attention degradation in extended contexts by systematically injecting structured semantic annotations into input documents.

The TAG framework operates through a two-stage process. As illustrated in Figure 2, our pipeline starts by segmenting the input text into a list of multi-sentence chunks, using a configurable chunking strategy and granularity. While the simplest strategy involves sentence- or paragraph-level chunking, more advanced techniques such as semantic chunking can also be applied. The resulting chunks are de-duplicated and passed into the respective tagging modules. The output is a list of annotated chunks with semantic tags, which can be used for downstream evaluation or to enhance generation tasks. Semantic elements are annotated using XML-style tags (e.g., `<sem_category>content</sem_category>`) that preserve document structure while providing explicit attention guidance during inference[2].

TAG is method-agnostic, supporting various tagging mechanisms including LLM-based approaches, traditional NLP tools and agentic systems. Unlike traditional retrieval-augmented generation methods that require external knowledge bases (Chen et al., 2025; Shen et al., 2025), TAG operates entirely within the input context, making it infrastructure-free and computationally efficient. The framework particularly excels in scenarios where relevant information exhibits minimal lexical overlap with queries, enabling stronger semantic reasoning across extended contexts without architectural modifications.

Semantic Tagging Methods - We apply semantic tagging to long-form documents using two approaches: with large language models (LLMs) and using traditional named entity recognition algorithms. Both methods operate over a shared preprocessing pipeline as depicted in Figure 2.

(i) LLM-based Tagging using privileged information - We explore two complementary strategies for semantic tagging using LLMs: Information Extraction (IE)-based tagging and classification-based tagging. Both approaches rely on carefully designed prompts that include persona setup, task instructions, output formatting constraints, and a predefined list of semantic tag categories with their definitions and examples.

In the IE-based approach, the LLM is prompted to extract entities from the input text and assign each to one of the defined semantic categories. Each entity is subsequently annotated in the input using a consistent text-based format, such as `<Person>Marie Curie</Person>`. These tags are inserted directly into the input and retained during inference, enabling the model to more effectively attend to semantically relevant spans when performing tasks such as question answering. A key advantage of this approach is that it eliminates the need for structured output formatting or post-processing such as output parsing. However, it may compromise input fidelity, as the model can hallucinate tags or alter the original text.

In the classification-based approach, we treat each semantic tag category as a separate class, and the prompt instructs the model to identify and output all tags that match the input text. The main

[2] The idea behind using XML-based tags to segment context has also been backed by a concurrent and independent work by Anthropic on context engineering.

advantage of this approach, compared to the IE prompting strategy, is the ability to support few-shot prompting, faster speed (owing to reduced maximum output token generation parameter), and no risk of novel tags or input text changes. The primary trade-offs are the need to enforce a specific output format within the prompt and to implement post-processing logic to extract the identified tags from the model's response.

To combine the strengths of both methods, we merge their outputs into a unified set of tagged chunks in this work. This hybrid strategy improves overall tag coverage and robustness, while mitigating the limitations of either method in isolation. The system prompts for both IE- and classification-based approaches are shown in Appendix A.4.3.

(ii) Traditional NER Tagging - An easy to implement alternative to LLM-based tagging is using Named Entity Recognition (NER) algorithms to extract entities and subsequently tag chunked text. In this work, we use spaCy (Honnibal et al., 2020), an industrial-strength natural language processing library, to perform NER on segmented text data. This model identifies 18 standard entity types, including persons, organizations, and geopolitical entities, etc (see Figure 10). The implementation employs a nested tagging approach, where text segments containing multiple entity types are wrapped with corresponding XML-style tags, preserving the hierarchical nature of entity relationships. The spaCy tagging approach ensures comprehensive entity coverage while maintaining the contextual integrity of the source materials.

Semantically enhancing documents and benchmarks using TAG - As discussed in Section 2, NIAH and complex narrative benchmarks provide a crucial test bed for revealing the limitations of modern LLMs. While such limits have been discovered and documented, it is helpful to enhance the test beds with universally applicable, practical alterations that can make the benchmark tasks easier, helping some LLMs and LLM-based agentic solutions gain performance. Towards this end, we demonstrate the applicability of TAG by proposing NoLiMa+ and NovelQA+, the tag-enriched versions of NoLiMa (Modarressi et al., 2025) and NovelQA (Wang et al., 2025) benchmarks, respectively. We note that TAG is not limited to these benchmarks and can be used to semantically improve any type of document using tags.

NoLiMa+ augments NoLiMa's test pairs by marking up haystack chunks that match candidate semantic tag categories. Each semantic tag is associated with definitions and examples, and chunks can receive multiple markups when appropriate. As mentioned in the original paper (Modarressi et al., 2025), the needles are defined using a list of keywords. For our paper, we use these keywords to define our list of privileged tags. We refer to Figure 9 for a complete list of tags and how they are defined in the prompts.

NovelQA+ enhances the NovelQA benchmark by applying semantic tagging to the novel texts used for question answering. Unlike NoLiMa+, which focuses on entity-specific tags relevant to needle-in-haystack retrieval, NovelQA+ employs broader narrative-oriented semantic categories to support complex reasoning across interconnected story elements. We segment each novel into manageable chunks and apply our TAG framework. NovelQA+ enables evaluation of how semantic tagging affects different types of reasoning complexity, from single-hop character identification to multi-hop plot analysis, providing insights into TAG's effectiveness across varying narrative comprehension challenges. In lieu of privileged information, we tag this benchmark using the spaCy entities. The complete list of tags is present in Figure 11.

4 Experiments

To comprehensively evaluate the effectiveness of TAG across different long-context scenarios, we conduct two sets of experiments – (i) long-context needle-in-haystack evaluation using the proposed NoLiMa+ benchmark (Section 3 and Section A.2.1), which tests semantic retrieval capabilities in the absence of lexical cues and (ii) complex narrative comprehension evaluation using the proposed NovelQA+ benchmark, which assesses reasoning over interconnected story elements across full-length novels (see Section 3 and Section A.2.2). As many of the novels exceeded the context window limit of Claude models, we had to truncate the books, and filter out questions that were from the omitted section. This left us with 1,035 questions ranging over different levels of complexity (single-hop, multi-hop, and detailed reasoning).

Through our experiments, we investigate the following research questions: **RQ1**: Can semantic tagging improve LLM performance on long-context retrieval tasks without relying on greedy search-based methods or architectural modifications? **RQ2**: How do different tagging methodolo-

Models	Tagged Context	Tagger	Tag definition in prompt	Context Length (CL)				Extremum drop rate
				250	500	16K	32K	
Claude 3.5 Sonnet	No	-	No	81.19	86.19	45.96	32.67	59.77
	No	-	Yes	**91.34**	**90.58**	**50.25**	36.45	60.1
	Yes	spaCy	Yes	88.77	88.36	47.65	34.63	60.99
	Yes	Privileged	Yes	87.53	85.35	48.4	**38.5**	56.0
Claude 3.7 Sonnet	No	-	No	94.66	93.48	55.76	45.56	51.87
	No	-	Yes	**97.12**	**95.7**	59.22	49.93	48.59
	Yes	spaCy	Yes	95.11	94.52	60.39	47.88	49.66
	Yes	Privileged	Yes	95.21	94.77	**60.78**	**52.39**	44.98

Table 1: Evaluation results on the NoLiMa+ benchmark. We show the accuracy scores of two Anthropic models, without (baseline) and with assistance via tags, across both the context, and the system prompt. The best scores for a configuration are shown in **bold**, while the second best are underlined. We also show the drop rates of each model from the 250 to 32K context lengths. Baseline results are shown in gray, improvements over baseline are highlighted in green and performance below baseline is shown in red.

gies, such as providing tag definitions alone versus augmenting the context with explicit tags, impact long-context performance and degradation rates?
RQ3: Does tagging help both non-reasoning and reasoning model performance?

Model Setup - For generating responses to questions based on untagged and tagged data, we consider two recent models from the Anthropic Claude families – Claude Sonnet 3.5 v2 (Anthropic, 2024) and Claude Sonnet 3.7 (Anthropic, 2025). Each of these models support a long context window of 200K tokens, thereby making them suitable for NIAH tasks such as NoLiMa and NovelQA. Additionally, these models rank among the best in terms of performance on long-query (i.e. >500 tokens) results in the LLMArena leaderboard (Chiang et al., 2024). We invoke both these models through Amazon Bedrock, integrating them directly into our codebase for automated inference over long documents. For Claude 3.5 Sonnet, we keep `temperature = 0` to make it as deterministic as possible in its prediction, while for Claude 3.7 Sonnet, we enable the `thinking` feature, which supersedes the choice of temperature.

Evaluation Metrics - For a fair comparison with the NoLiMa benchmark, we adopt the same evaluation metrics. More specifically, we give an LLM a score of 1 as long as the generated response *contains* the golden answer, else 0. Furthermore, we conduct our experiments using context lengths of 250, 500, 16K and 32K tokens respectively[3].

The evaluation methodology for NovelQA follows a straightforward multiple-choice assessment framework. We evaluate the model performance using *exact match* accuracy by comparing the LLM's single-character output (A, B, C, or D) against the golden answer provided in the NovelQA dataset.

Experimental settings - We evaluated two semantic tagging approaches for the long-context understanding tasks: **(i) Tag Definitions only (TD)**: We hypothesize that providing explicit definitions of key semantic elements within the task prompt can enable models to better identify relevant information across increasing context lengths. **(ii) Tag Definitions with Tagged Context (TD+TC)**: Our second experimental setting involves combining tag definitions in the prompt with guided context navigation using tagged context. By strategically applying semantic tags throughout the context and providing explicit definitions of these tags, we expect models to maintain stronger associative reasoning capabilities even at the longest context lengths. The baseline model in our experiments is the vanilla RAG setting with NO tagged context, and NO tag definitions in prompt.

Evaluation Results and Discussion - Table 1 presents the accuracy scores across various context lengths ranging from 250 to 32K tokens. In the baseline condition (i.e., no semantic tagging and no tag definition in prompt), Claude 3.5 Sonnet's performance dropped from 81.19% at CL250 to 32.67% at CL32K, representing a 59.77% extremum drop rate. Similarly, Claude 3.7 Sonnet experienced a 51.87% extremum drop rate from 94.66% at CL250 to 45.56% at CL32K.

The TD approach improved short-context performance for both models compared to the baseline, with Claude 3.5 Sonnet reaching 91.34% at CL250

[3] Our approach is not limited by the context length, but we want to emphasize the utility of tagging really long contexts like 16K and 32K tokens, by benchmarking them against shorter contexts of token sizes 250 and 500.

Models	Tagged Context	Tagger	Tag definition in prompt	Complexity		
				Single-hop	Multi-hop	Detail
Claude 3.5 Sonnet	No	-	No	86.88	48.71	73.36
	No	-	Yes	86.39	**50.12**	74.77
	Yes	spaCy	Yes	**87.87**	49.88	**78.97**
Claude 3.7 Sonnet	No	-	No	90.10	56.72	**84.11***
	No	-	Yes	90.84	56.23	82.71
	Yes	spaCy	Yes	**91.09***	**56.97***	81.31

Table 2: Evaluation results on the NovelQA+ benchmark. We show the accuracy scores of two Anthropic models, without (baseline) and with assistance via tags, across both the context and the system prompt. The best scores are shown in **bold**, while the second best are underlined. Overall highest accuracy is marked with an asterisk.

(+10.15 gain) and Claude 3.7 Sonnet reaching 97.12% (+2.46 gain). When incorporating SpaCy tagging, the models achieved 88.77% and 95.11% at CL250 for Claude 3.5 and 3.7 respectively. This demonstrates that presence of clear task definitions in the prompt could significantly enhance models' ability to establish semantic connections (**RQ2**). However, this approach did not substantially reduce the extremum drop rate, with Claude 3.5 Sonnet still experiencing a 60.1% decline, SpaCy showing a 60.99% decline, and Claude 3.7 Sonnet a 48.59% decline from peak performance to CL32K.

While the privileged tagging approach (TD + TC) showed slightly lower peak performance compared to TD alone for CL250 (87.53% vs 91.34% for Claude 3.5, 95.21% vs 97.12% for Claude 3.7), it significantly reduced degradation at CL32K, with Claude Sonnet 3.7 achieving 52.39% performance accuracy, compared to 49.93% with TD alone and 45.56% in the baseline (**RQ1**). This suggests that tagging becomes increasingly effective as context length extends beyond certain thresholds. Figure 12 shows an illustration of Claude 3.7's ability to reason about the tags present in the context to accurately extract the correct answer.

Table 2 presents the evaluation results for the NovelQA dataset. The experimental analysis reveals distinct patterns across different model configurations and complexity categories. In the baseline configuration (Section A.4.1), Claude 3.7 Sonnet demonstrates superior performance compared to 3.5 Sonnet v2 across all complexity categories (90.10% vs 86.88% for single-hop, 56.72% vs 48.17% for multi-hop, and 84.11% vs 73.36% for detailed questions). With tag definitions only (TD) (Figure 5), both models show modest improvements in certain categories, with Claude 3.7 achieving 90.84% for single-hop questions and Claude 3.5 Sonnet showing improved performance in multi-hop scenarios (50.12% vs baseline 48.17%). The integration of TD+TC with spaCy (Figure 6) yields the most promising results, particularly for Claude 3.7 Sonnet, which achieves the highest accuracy across single-hop (91.09%) and multi-hop (56.97%) scenarios. Notably, Claude 3.5 Sonnet v2 shows substantial improvement in detailed questions under this configuration, reaching 78.97% accuracy compared to its baseline of 73.36%. These results suggest that the combination of TD+TC can enhance model performance, particularly for more complex reasoning tasks. Considering the above observations in both experiments, semantic tagging using TAG improves the performance of LLMs in both non-reasoning and reasoning settings (**RQ3**).

5 Conclusion

We proposed TAG, a lightweight prompting and input augmentation technique that explicitly highlights salient elements in long and complex contexts through structured semantic annotations. By injecting XML-style tags directly into input documents, our method helps models focus their attention during inference without requiring architectural modifications or retrieval infrastructure. TAG is method-agnostic, supporting various tagging mechanisms including LLM-based approaches, traditional NLP tools, and agentic systems, making it infrastructure-free and computationally efficient.

Through extensive experiments on NoLiMa+ and NovelQA+ benchmarks, we demonstrate that semantic tagging improves question-answering accuracy by over 17% for 32K token contexts and 2.9% in complex reasoning tasks, while significantly reducing performance degradation rates. Our results establish TAG as a practical and scalable approach for strengthening LLM performance in extended inference scenarios without the computational overhead of traditional retrieval systems.

Limitations

While TAG shows promising results, several areas warrant further investigation. Our evaluation focuses on question-answering tasks within synthetic (NoLiMa+) and literary (NovelQA+) domains; broader validation across diverse tasks and technical domains would strengthen our findings. The framework's effectiveness depends on appropriate semantic category design, which may present challenges in specialized domains or low-resource languages.

TAG introduces modest computational overhead during preprocessing, though this remains lighter than traditional retrieval systems. While our method enhances attention to relevant content, it operates within existing context windows and cannot address fundamental memory limitations of current LLMs. We note that our experiments primarily utilized spaCy-based and LLM-based tagging; comprehensive evaluation of agentic tagging approaches on-the-fly at inference presents an important direction for future work.

References

Anthropic. 2024. Claude 3.5 sonnet model card addendum.

Anthropic. 2025. Claude 3.7 sonnet system card.

Peter Baile Chen, Tomer Wolfson, Michael Cafarella, and Dan Roth. 2025. Enrichindex: Using llms to enrich retrieval indices offline. *Preprint*, arXiv:2504.03598.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *Preprint*, arXiv:2402.13753.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. *bibsonomy*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *Preprint*, arXiv:2401.01325.

Gregory Kamradt. 2023. Needle in a haystack - pressure testing llms. *GitHub*.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schutze. 2025. Nolima: Long-context evaluation beyond literal matching. *ArXiv*, abs/2502.05167.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *Preprint*, arXiv:2202.08904.

Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. 2024. Tag-llm: Repurposing general-purpose llms for specialized domains. *Preprint*, arXiv:2402.05140.

Wenxuan Shen, Mingjia Wang, Yaochen Wang, Dongping Chen, Junjie Yang, Yao Wan, and Weiwei Lin. 2025. Are we on the right way for assessing document retrieval-augmented generation? *Preprint*, arXiv:2508.03644.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context? *Preprint*, arXiv:2109.09115.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Xiangkun Hu, Zheng Zhang, Qian Wang, and Yue Zhang. 2025. Novelqa: Benchmarking question answering on documents exceeding 200k tokens. *Preprint*, arXiv:2403.12766.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2024. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Preprint*, arXiv:2403.04797.

A Appendix

A.1 Background and Related Work

A.1.1 Long-Context Challenges in LLMs

Despite the advancements enabling LLMs to process extended sequences, empirical evidence indicates persistent challenges in effectively utilizing long contexts. Several factors contribute to LLM failures in long-context settings, including attention degradation, lost-in-the-middle and literal match dependence. We briefly describe these factors below.

Attention Degradation. The softmax-based self-attention mechanism in Transformer architectures struggle to maintain focus as context length increases. This causes diluted attention weights, particularly for the tokens in the middle of the input sequences (Liu et al., 2023).

Lost-in-the-Middle. This phenomenon describes a systematic attention bias where language models struggle to accurately retrieve and utilize information positioned in the middle portions of long contexts. Recent empirical studies have demonstrated that tokens located centrally within extensive sequences receive disproportionately less attention compared to those at the beginning and end (Liu et al., 2023). This positional bias leads to a significantly reduced performance for content placed in middle segments. The effect becomes increasingly evident as context length expands (Zhang et al., 2024).

Literal Match Dependence. LLMs often demonstrate dependence on literal lexical overlap, succeeding primary when queries and relevant passages share explicit term matches, which limits their ability to generalize to semantically similar, but lexically distinct contexts (Modarressi et al., 2025; Sun et al., 2021). Particularly, the NoLiMa benchmark (Modarressi et al., 2025) illustrates these issues by removing literal cues and requiring associative reasoning. Even top-tier models like GPT-4o drop from 99.3% accuracy at 1K tokens to 69.7% at 32K tokens on NoLiMa.

A.1.2 Input Tagging and Modular Prompt Conditioning

Our work is closely related to that of Tag-LLM (Shen et al., 2024), which introduces a framework for repurposing general-purpose language models for specialized domains via learnable input tags. These tags are continuous embeddings prepended to the model's input sequence to encode domain- or task-specific information. Tag-LLM learns two types of tags, i.e. domain tags and function tags, through a hierarchical protocol that leverages both unlabeled and labeled data. Notably, Tag-LLM retains these embedding-level tags during inference, conditioning the model throughout the input sequence. In contrast, our semantic tagging approach operates purely at the textual level, using LLM-generated markers to highlight salient content. While both methods retain tags during inference to guide model behavior, Tag-LLM relies on fine-tuned internal conditioning, whereas our approach offers a lightweight, interpretable alternative that requires no model modifications or additional training.

A.2 Datasets

A.2.1 NoLiMa

Our experiments utilize the NoLiMa (No Literal Matching) benchmark (Modarressi et al., 2025), a recently developed evaluation framework specifically designed to assess semantic understanding capabilities of LLMs in long-context scenarios. Unlike traditional long-context benchmarks that rely on lexical overlap between questions and relevant content, NoLiMa deliberately minimizes literal matches, requiring models to make latent semantic connections to succeed. Each evaluation instance consists of a 'needle' (i.e., a sentence containing key information about a character and a concept), a 'question' that refers to the needle using semantically related but lexically distinct keywords and a 'haystack' of irrelevant text. For example, a needle might state 'Yuki lives next to the Semper Opera House' while the question asks 'Which character has been to Dresden?', requiring the model to recognize the semantic association between 'Semper Opera House' and 'Dresden' without surface-level matching.

This absence of literal matching creates a particularly challenging test case for our research, as it forces models to rely on their deeper semantic reasoning rather than pattern matching abilities. The benchmark comprises 58 needle-question pairs with varying complexity levels, including both one-hop and two-hop reasoning paths. To evaluate the performance across different context lengths, each needle is placed at 26 fixed positions throughout haystacks exceeding 60K tokens, constructed from concatenated snippets of open-licensed books. The benchmark's design specifically addresses limitations in existing evaluation frameworks that inadvertently allow models to succeed through surface-level cues rather than true semantic understanding, making it ideal for evaluating our semantic tagging approach.

A.2.2 NovelQA

NovelQA is a benchmark dataset for evaluating long-text language models' ability to understand and answer question about novels. Distinguished by its manually annotated questions crafted by domain experts, the dataset comprises 2,305 questions spanning 89 novels, with a robust taxonomy classifying queries by both complexity (multi-hop, single-hop, and detail) and aspect (times, meaning, span, setting, relation, character, and plot). The benchmark's architecture facilitates comprehensive evaluation through both generative and multiple-choice paradigms, enabling granular assessment of models' capabilities in long-range comprehension and reasoning. The NovelQA has the rigorous methodology in dataset construction and evaluation. The questions are deliberately designed to test various of text comprehension while maintaining high-quality control through expert annotation and validation. This framework not only provides quantitative metrics for model performance but also offers insights into specific challenges in long-context understanding, such as evidence position effects and information recall capabilities. The benchmark thus serves as a crucial tool for advancing our understanding of language models' limitations and capabilities in processing extended narratives, contributing significantly to the development of more robust long-context language models.

A.3 NoLiMa+ Dataset Creation Pipeline

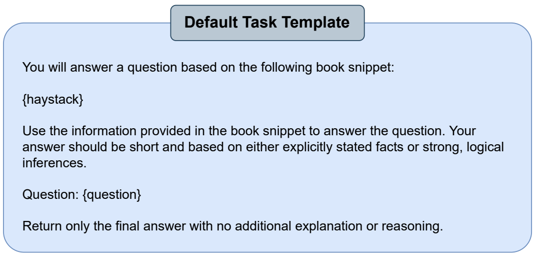
The NoLiMa+ dataset creation process involves a systematic multi-stage pipeline designed to enhance the original NoLiMa benchmark with semantic tagging capabilities. The enhanced dataset enables more targeted evaluation of retrieval-augmented generation systems by providing structured semantic annotations that guide attention to relevant content sections. We start by segmenting each haystack into a list of multi-sentence chunks, using a predefined chunking strategy and granularity, which is defined as the 'maximum chunk size'.The simplest chunking strategy is so-called sentence or paragraph chunking, but it could easily be any other chunking technique, e.g. semantic chunking.

The list of chunks is then de-duplicated, and each chunk is passed to two complementary LLM-based tagging sub-pipelines. The key difference between these is their prompting strategy. In addition to the chunk text, each of the two sub-pipelines' prompts also includes persona, task and formatting instructions, and a list of the candidate semantic tag categories, their definitions, and examples.

After output parsing and post-processing steps, the outputs of the two sub-pipelines are merged into a single tagged chunk. All tagged chunks are concatenated to derive the tagged haystacks, which can be then be used to benchmark Tagging-Augmented Generation of arbitrary LLMs or AI Agents.

A.4 Prompts Used

A.4.1 User prompts



Default Task Template

You will answer a question based on the following book snippet:

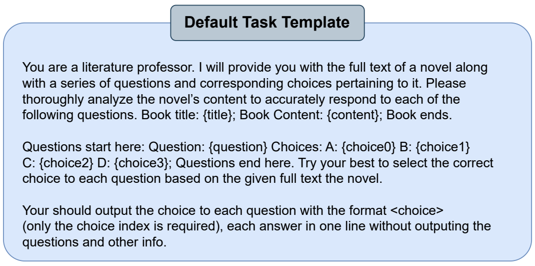
{haystack}

Use the information provided in the book snippet to answer the question. Your answer should be short and based on either explicitly stated facts or strong, logical inferences.

Question: {question}

Return only the final answer with no additional explanation or reasoning.

Figure 3: User prompt for NoLima+ dataset.



Default Task Template

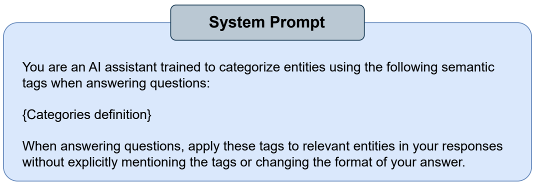
You are a literature professor. I will provide you with the full text of a novel along with a series of questions and corresponding choices pertaining to it. Please thoroughly analyze the novel's content to accurately respond to each of the following questions. Book title: {title}; Book Content: {content}; Book ends.

Questions start here: Question: {question} Choices: A: {choice0} B: {choice1} C: {choice2} D: {choice3}; Questions end here. Try your best to select the correct choice to each question based on the given full text the novel.

Your should output the choice to each question with the format <choice> (only the choice index is required), each answer in one line without outputing the questions and other info.

Figure 4: User prompt for NovelQA+ dataset.

A.4.2 System prompts for TAG framework



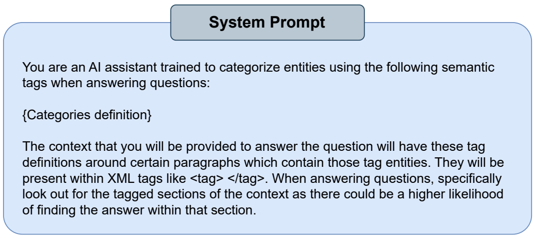
System Prompt

You are an AI assistant trained to categorize entities using the following semantic tags when answering questions:

{Categories definition}

When answering questions, apply these tags to relevant entities in your responses without explicitly mentioning the tags or changing the format of your answer.

Figure 5: System prompt for untagged context with tag definition in prompt.



System Prompt

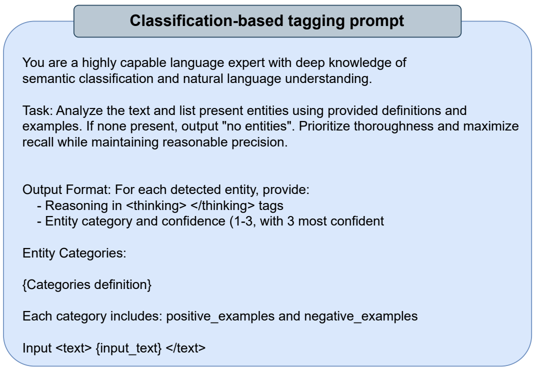
You are an AI assistant trained to categorize entities using the following semantic tags when answering questions:

{Categories definition}

The context that you will be provided to answer the question will have these tag definitions around certain paragraphs which contain those tag entities. They will be present within XML tags like <tag> </tag>. When answering questions, specifically look out for the tagged sections of the context as there could be a higher likelihood of finding the answer within that section.

Figure 6: System prompt for tagged context with tag definition in prompt.

A.4.3 Semantic tagging prompts



Classification-based tagging prompt

You are a highly capable language expert with deep knowledge of semantic classification and natural language understanding.

Task: Analyze the text and list present entities using provided definitions and examples. If none present, output "no entities". Prioritize thoroughness and maximize recall while maintaining reasonable precision.

Output Format: For each detected entity, provide:
- Reasoning in <thinking> </thinking> tags
- Entity category and confidence (1-3, with 3 most confident

Entity Categories:

{Categories definition}

Each category includes: positive_examples and negative_examples

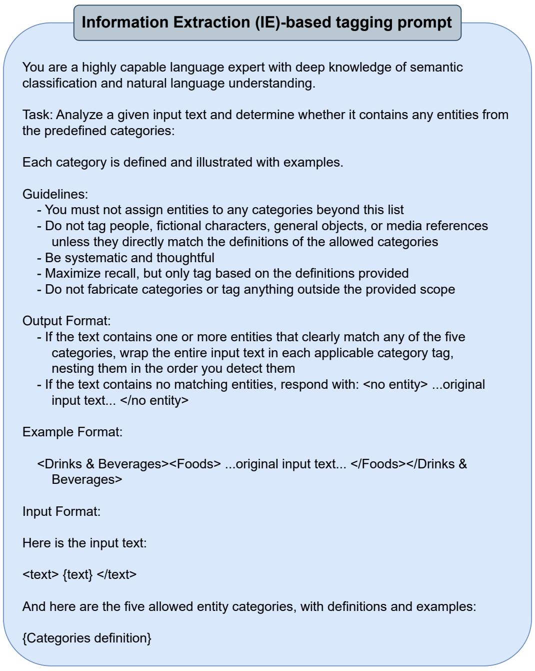
Input <text> {input_text} </text>

Figure 7: Classification-based tagging prompt.

A.4.4 Tag category definitions

spaCy provides a list of default named entities which can be mapped to semantic categories. These are shown in Figure 10.

A.5 Illustration of TAG helping LLM reasoning



Information Extraction (IE)-based tagging prompt

You are a highly capable language expert with deep knowledge of semantic classification and natural language understanding.

Task: Analyze a given input text and determine whether it contains any entities from the predefined categories:

Each category is defined and illustrated with examples.

Guidelines:
- You must not assign entities to any categories beyond this list
- Do not tag people, fictional characters, general objects, or media references unless they directly match the definitions of the allowed categories
- Be systematic and thoughtful
- Maximize recall, but only tag based on the definitions provided
- Do not fabricate categories or tag anything outside the provided scope

Output Format:
- If the text contains one or more entities that clearly match any of the five categories, wrap the entire input text in each applicable category tag, nesting them in the order you detect them
- If the text contains no matching entities, respond with: <no entity> ...original input text... </no entity>

Example Format:

<Drinks & Beverages><Foods> ...original input text... </Foods></Drinks & Beverages>

Input Format:

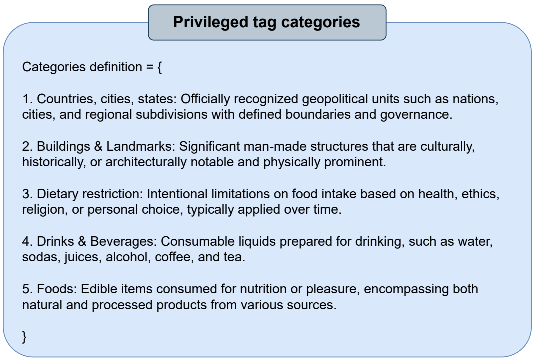
Here is the input text:

<text> {text} </text>

And here are the five allowed entity categories, with definitions and examples:

{Categories definition}

Figure 8: Information Extraction (IE)-based tagging prompt.



Privileged tag categories

Categories definition = {

1. Countries, cities, states: Officially recognized geopolitical units such as nations, cities, and regional subdivisions with defined boundaries and governance.

2. Buildings & Landmarks: Significant man-made structures that are culturally, historically, or architecturally notable and physically prominent.

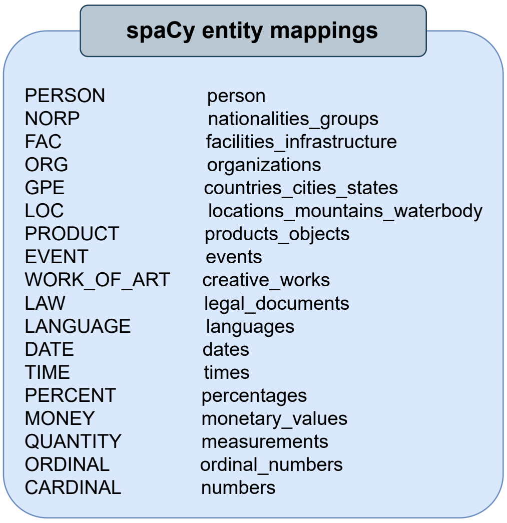
3. Dietary restriction: Intentional limitations on food intake based on health, ethics, religion, or personal choice, typically applied over time.

4. Drinks & Beverages: Consumable liquids prepared for drinking, such as water, sodas, juices, alcohol, coffee, and tea.

5. Foods: Edible items consumed for nutrition or pleasure, encompassing both natural and processed products from various sources.

}

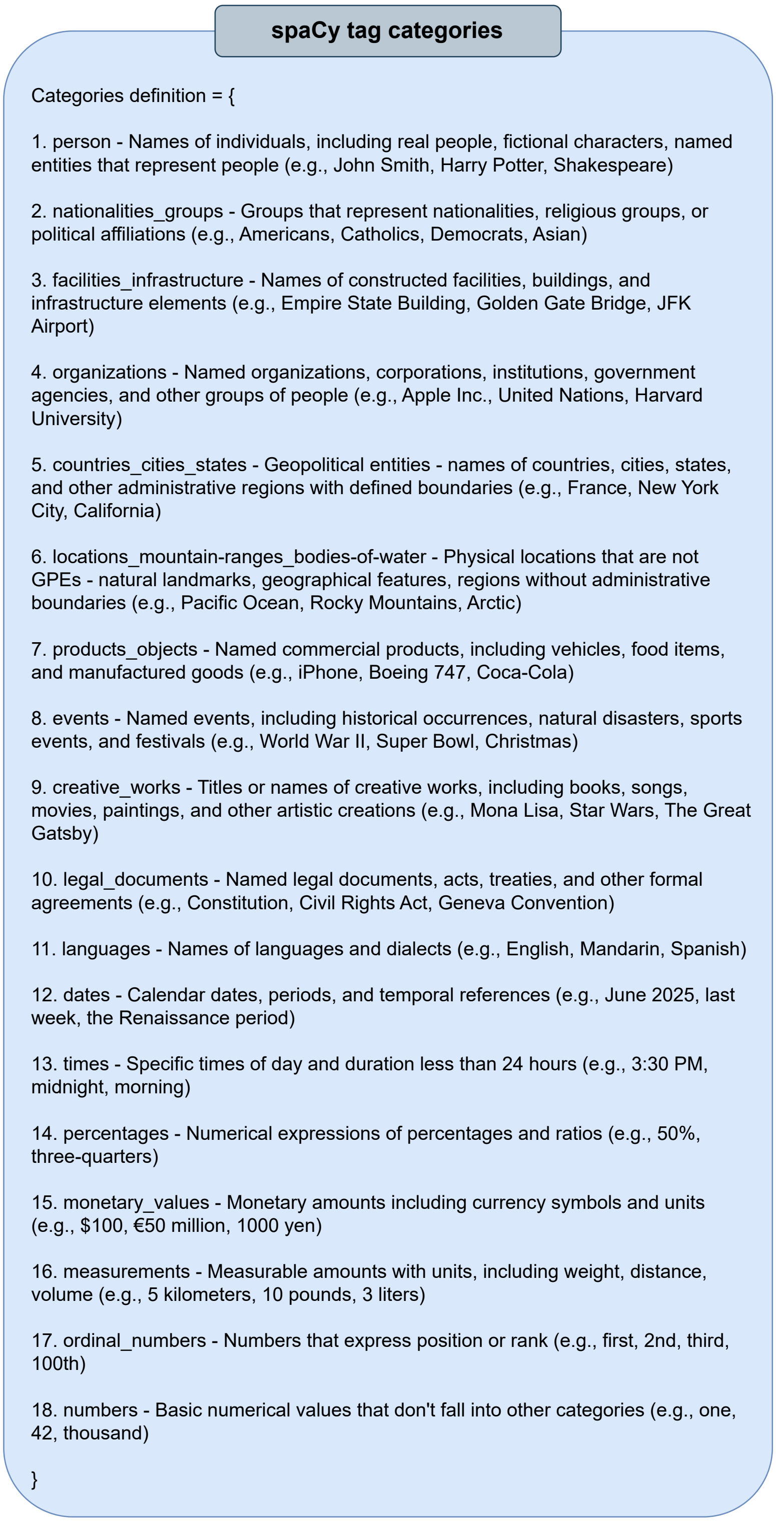
Figure 9: Privileged tag categories.



spaCy entity mappings

PERSON	person
NORP	nationalities_groups
FAC	facilities_infrastructure
ORG	organizations
GPE	countries_cities_states
LOC	locations_mountains_waterbody
PRODUCT	products_objects
EVENT	events
WORK_OF_ART	creative_works
LAW	legal_documents
LANGUAGE	languages
DATE	dates
TIME	times
PERCENT	percentages
MONEY	monetary_values
QUANTITY	measurements
ORDINAL	ordinal_numbers
CARDINAL	numbers

Figure 10: spaCy entity mappings.



spaCy tag categories

Categories definition = {

1. person - Names of individuals, including real people, fictional characters, named entities that represent people (e.g., John Smith, Harry Potter, Shakespeare)

2. nationalities_groups - Groups that represent nationalities, religious groups, or political affiliations (e.g., Americans, Catholics, Democrats, Asian)

3. facilities_infrastructure - Names of constructed facilities, buildings, and infrastructure elements (e.g., Empire State Building, Golden Gate Bridge, JFK Airport)

4. organizations - Named organizations, corporations, institutions, government agencies, and other groups of people (e.g., Apple Inc., United Nations, Harvard University)

5. countries_cities_states - Geopolitical entities - names of countries, cities, states, and other administrative regions with defined boundaries (e.g., France, New York City, California)

6. locations_mountain-ranges_bodies-of-water - Physical locations that are not GPEs - natural landmarks, geographical features, regions without administrative boundaries (e.g., Pacific Ocean, Rocky Mountains, Arctic)

7. products_objects - Named commercial products, including vehicles, food items, and manufactured goods (e.g., iPhone, Boeing 747, Coca-Cola)

8. events - Named events, including historical occurrences, natural disasters, sports events, and festivals (e.g., World War II, Super Bowl, Christmas)

9. creative_works - Titles or names of creative works, including books, songs, movies, paintings, and other artistic creations (e.g., Mona Lisa, Star Wars, The Great Gatsby)

10. legal_documents - Named legal documents, acts, treaties, and other formal agreements (e.g., Constitution, Civil Rights Act, Geneva Convention)

11. languages - Names of languages and dialects (e.g., English, Mandarin, Spanish)

12. dates - Calendar dates, periods, and temporal references (e.g., June 2025, last week, the Renaissance period)

13. times - Specific times of day and duration less than 24 hours (e.g., 3:30 PM, midnight, morning)

14. percentages - Numerical expressions of percentages and ratios (e.g., 50%, three-quarters)

15. monetary_values - Monetary amounts including currency symbols and units (e.g., $100, €50 million, 1000 yen)

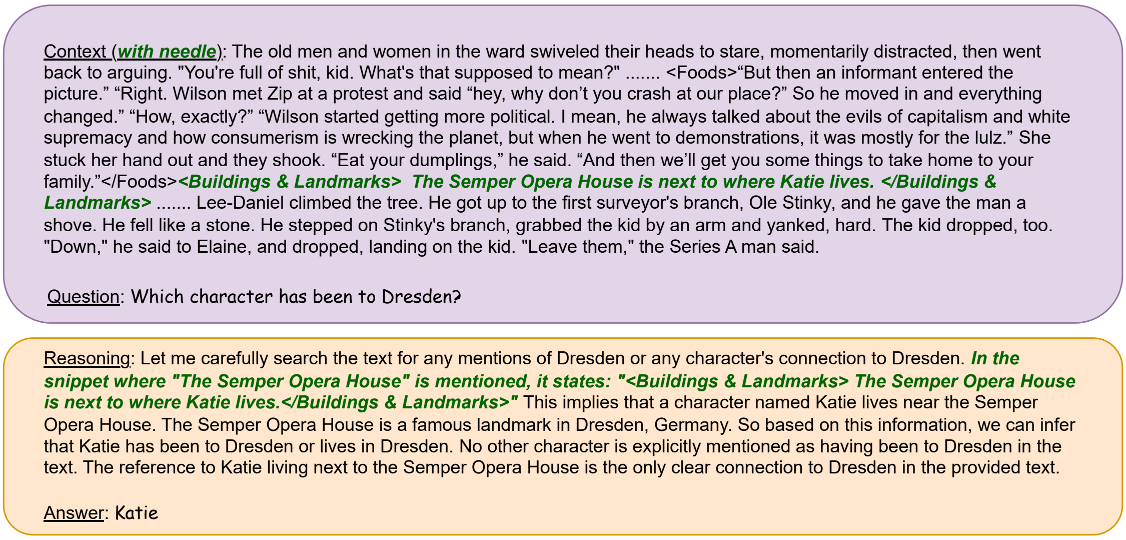
16. measurements - Measurable amounts with units, including weight, distance, volume (e.g., 5 kilometers, 10 pounds, 3 liters)

17. ordinal_numbers - Numbers that express position or rank (e.g., first, 2nd, third, 100th)

18. numbers - Basic numerical values that don't fall into other categories (e.g., one, 42, thousand)

}

Figure 11: spaCy tag categories.



Context (***with needle***): The old men and women in the ward swiveled their heads to stare, momentarily distracted, then went back to arguing. "You're full of shit, kid. What's that supposed to mean?" .... <Foods>“But then an informant entered the picture.” “Right. Wilson met Zip at a protest and said “hey, why don’t you crash at our place?” So he moved in and everything changed.” “How, exactly?” “Wilson started getting more political. I mean, he always talked about the evils of capitalism and white supremacy and how consumerism is wrecking the planet, but when he went to demonstrations, it was mostly for the lulz.” She stuck her hand out and they shook. “Eat your dumplings,” he said. “And then we’ll get you some things to take home to your family.”</Foods>***<Buildings & Landmarks> The Semper Opera House is next to where Katie lives. </Buildings & Landmarks>*** .... Lee-Daniel climbed the tree. He got up to the first surveyor's branch, Ole Stinky, and he gave the man a shove. He fell like a stone. He stepped on Stinky's branch, grabbed the kid by an arm and yanked, hard. The kid dropped, too. "Down," he said to Elaine, and dropped, landing on the kid. "Leave them," the Series A man said.

Question: Which character has been to Dresden?

Reasoning: Let me carefully search the text for any mentions of Dresden or any character's connection to Dresden. ***In the snippet where "The Semper Opera House" is mentioned, it states: "<Buildings & Landmarks> The Semper Opera House is next to where Katie lives.</Buildings & Landmarks>"*** This implies that a character named Katie lives near the Semper Opera House. The Semper Opera House is a famous landmark in Dresden, Germany. So based on this information, we can infer that Katie has been to Dresden or lives in Dresden. No other character is explicitly mentioned as having been to Dresden in the text. The reference to Katie living next to the Semper Opera House is the only clear connection to Dresden in the provided text.

Answer: Katie

Figure 12: [Best viewed in color] Illustration of the benefits of tagging context in generating correct responses by Claude 3.7 Sonnet. In the above example, a long context with an embedded needle sentence has been tagged by our proposed approach. As seen in the generating reasoning statement, this helps the LLM to attend to the correct part of the context to easily extract the answer to an otherwise difficult question.