Evaluating and Predicting the Performance of Large Language Models on Long-Context Tasks

Yike Wang

Acknowledgement

I’m deeply grateful to Kevin Lin for his mentorship and contributions to this project, as well as to Professor Joseph E. Gonzalez and Professor Daniel Klein for their invaluable guidance. Working with all of you has been an incredible opportunity, and I truly appreciate it.

Evaluating and Predicting the Performance of Large Language Models on Long-Context Tasks

by Yike Wang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Joseph E. Gonzalez

Professor Joseph E. Gonzalez
Research Advisor

May 15, 2024

(Date)

Dan Klein

Professor Daniel Klein
Second Reader

May 17, 2024

(Date)

Evaluating and Predicting the Performance of Large Language Models on Long-Context Tasks

by

Yike Wang

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Joseph E. Gonzalez, Chair
Professor Daniel Klein

Spring 2024

Evaluating and Predicting the Performance of Large Language Models on Long-Context Tasks

Abstract

Evaluating and Predicting the Performance of Large Language Models on Long-Context Tasks

by

Yike Wang

Master of Science in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Joseph E. Gonzalez, Chair

Under certain circumstances, we may lack the computational resources and data necessary to conduct experiments and assess the performance of a specific model on a given long-context task. In this work, we explore the potential factors associated with long-context performance by studying the correlation between different unit tasks and different context lengths. We propose a new multi-context evaluation dataset with three distinct tasks based on context accessing patterns, featuring adjustable context length, automated evaluation, and diagnosis beyond overall accuracy. All unit tasks present unexpected challenges for state-of-the-art large language models, and we observe a significant decrease in performance as token length increases. We hypothesize that, in terms of long-context prediction, the type of task provides more informative insights than context length and assessing the long-context understanding of language models solely through the needle-in-the-haystack approach [18] lacks informativeness.

Contents

Acknowledgments

I’m deeply grateful to Kevin Lin for his mentorship and contributions to this project, as well as to Professor Joseph E. Gonzalez and Professor Daniel Klein for their invaluable guidance. Working with all of you has been an incredible opportunity, and I truly appreciate it.

Chapter 1

Introduction

Large language models (LLMs) have demonstrated impressive capabilities across many NLP tasks [7, 23]. However, they still suffers from limited context window, making them incapable of solving NLP tasks beyond this limit. Prior research have made significant contributions on extending the context length of LLMs, including position interpolation [9, 24, 10] and novel training methods that allow access to longer context [11, 22].

However, there are instances where we may lack the computational resources and data necessary to conduct experiments and assess the performance of a specific model on a given long-context task. This leads us to question how we can *predict the long-context performance of language models*. One potential subsequent query could be: Which provides more informative insights - the model's performance on the same task with a shorter context or its performance on different tasks with similar context lengths?

Existing evaluation benchmarks for long-context understanding primarily rely on the performance of LLMs on downstream tasks such as summarization and code completion [25, 6]. However, these tasks are complex in the way that they often involve various fundamental context understanding skills, making them hard to be used for prediction. Therefore, we introduce a new set of prompting-based long-context evaluation tasks grounded in context assessing patterns. We argue that the majority of downstream long-context tasks employ basic context assessing patterns including key retrieval, filtering by predicates, and information exclusion. Accordingly, we structure our evaluation tasks to reflect these patterns, aiming to provide straight insights into how effectively LLMs can engage with and interpret long contexts, with following key features:

- *Adjustable Context Length*: Each task within our evaluation dataset is designed to be adaptable in length, reaching up to 1.2 billion words, achieved by changing the number of context pieces. With that, our dataset can accommodate different context windows of different LLM.

- *Automated Evaluation*: Our evaluation dataset allows automated evaluation, without the involvement of other language models or human annotators, thereby reducing bias and subjectivity in the evaluation process.

- *Multi-Context*: The context pieces in our evaluation dataset is drawn from multiple passages spanning different knowledge domains, which challenges LLMs to comprehend and reason across diverse knowledge domains. This setup aligns with real-world scenarios where long-context use cases often involve sourcing from multiple passages.
- *Diagnosis Beyond Overall Accuracy*: Besides the overall accuracy, our dataset enables us to conduct diagnostic analyses across various dimensions such as position, selectivity, and proportion. This comprehensive approach offers deeper insights into model performance, allowing for a more nuanced understanding of its strengths and weaknesses in different contexts.

Through experiments with five state-of-the-art open and closed large language models, we observe a significant decrease in performance as token length increases. Furthermore, all unit tasks present unexpected challenges for these language models, with performance falling notably short of perfection. Further analysis reveals that retrieval quality remains steady across various key positions, indicating that language models may be trained to alleviate the ”lost-in-the-middle” phenomenon. Moreover, we note that as selectivity rises, filtering accuracy tends to decrease, although it remains consistent across different exclusion proportions.

Further qualitative analysis shows that as the context length increases beyond certain thresholds where models struggle to manage effectively, instead of precisely addressing the initial parts of the context, the model tends to fail entirely, leading to inaccuracies even in information located at the beginning. Synthetic experiments show that similar to natural languages, performance over discrete numbers is significantly influenced by context length, with most models struggling on basic aggregation tasks involving 500 or even 100 integers.

By studying the performance correlation between different tasks and different token lengths, we find that while different language models exhibit markedly different performances on the Key Retrieval task, their performance on the filtering task remains relatively consistent. Also, models that perform exceptionally with short contexts typically also perform considerably well in the same task with longer contexts. Therefore, we hypothesize that, in terms of long-context prediction, the type of task provides more informative insights than context length; performance variations are more evident across different tasks than across different context lengths. Building upon this observation, we posit that assessing the long-context understanding of language models solely through the needle-in-the-haystack approach [18], despite its adaptable context length, lacks informativeness. Developing and evaluating additional unit tasks are imperative to obtain a more precise estimation of language models’ performance on downstream long-context tasks.

Chapter 2

Related Work

2.1 Expanding the Context Window of Large Language Models

Currently, the standard context limit for large language models spans from 2,000 tokens to 32,000 tokens [26, 15, 2], with only a few being able to accommodate an impressive 128,000 tokens [23, 13]. However, when dealing with lengthy texts such as books or historical records, these window sizes are inadequate, hindering LLMs from effectively processing such data.

Researchers have thus explored various methods to overcome this limitation. One such approach is Transformer-XL [10], which enables learning dependency beyond a fixed length without disrupting temporal coherence by a segment-level recurrence mechanism and a novel positional encoding scheme. Another method, called YaRN [24], extends the context window of models trained with Rotary Position Embeddings by employing Dynamic Scaling. Meanwhile, Position Interpolation [9] is a technique that also extends the context window sizes of RoPE-based LLMs by linearly downscaling the input position indices to match the original context window size. Moreover, prior studies have also addressed the context limit challenge through dilated attention [11], which exponentially expands the attentive field as the distance increases, and landmark attention [22], which trains attention mechanisms to utilize a landmark token for selecting relevant blocks.

In this work, we open up a new research dimension on predicting the performance of long-context LLMs and explore the correlation between performance across various long-context tasks. We create a new evaluation dataset and investigate potential factors associated with long-context performance.

2.2 Long-Context Benchmarks

In the prior studies, the long-context evaluation primarily relies on the performance on downstream tasks [1, 27, 19, 20, 8]. ZeroSCROLLS [25] presents a zero-shot benchmark for long text understanding, including tasks such as determining the percentage of positive reviews and reordering the chapters according to chapter summaries. LongBench [6] is a bilingual benchmark featuring tasks like summarization and code completion, while BAMBOO [12] includes hallucination detection, text sorting, and code completions tasks.

The Needle-in-a-Haystack task [18], where a random fact or statement is placed in the middle of a long context window, and the model is asked to retrieve this statement, serves as the only evaluation of the basic in-context retrieval ability of long context LLMs. However, this task proves too easy for current powerful LLMs; Claude-3.0 is even able to acknowledge that the ”needle” is most likely not part of the original document, but a joke or a test [4]. Therefore, we propose more challenging long-context evaluation tasks based on context assessing patterns in this work. Similar to Needle-in-a-Haystack, our tasks are adaptable to varying context lengths, automatically evaluated, and can conduct diagnostics beyond overall accuracy metrics.

Chapter 3

Evaluation Dataset

3.1 Database

Our database, primarily sampled from RoMQA database [28], which includes Wikidata, T-REX, and crowd-source question annotations, contains a collection of 1,000 multi-constraint queries spanning various knowledge domains. For example, in the query ”I’m looking for academics who worked at Harvard and MIT and went to school at Harvard,” there are three distinct constraints: employer of Harvard, employer of MIT, and educated at Harvard. On average, each query contains 2.435 constraints. For every query, there are 100 potential candidates, along with several pieces of context validating each constraint mentioned within the query. In the example provided, the relevant context would include evidence on employment and education respectively. An example context on education will be ”Bart Andrew Kosko (born February 7, 1960) is a writer and professor of electrical engineering and law at the University of Southern California (USC).” The length of each context piece also varies, with an average of 87 words. The context length of each task can extend up to 1.2 billion words. The potential candidates may either fully, partially, or not at all meet the specified constraints within each query.

3.2 Unit Tasks

Using our database, we create a set of evaluation tasks based on different context accessing patterns. We argue that the performance of LLMs on these unit tasks is more informative and can build up for predicting performance on more complex long-context tasks. See details of unit tasks in Appendix A.1.

Key Retrieval The primary long-context access pattern is key retrieval, identifying all relevant information, or ”values”, within the extensive context according to the instructions, or ”keys”, of the downstream tasks. In this unit task, a list of candidates with context are given, and the model is instructed to find and return the context related to a certain candidate in the given list. One in-context exemplar is provided under this task.

Filter by Predicates The other major access pattern is filtering by predicates, selecting information from the context by certain predicates as instructed. For instance, in the summarization task, the model must select information related to the main ideas of the document instead of minor details. In our unit task, the model is presented with a multi-constraint query and a list of potential candidates, each accompanied by multiple context pieces on queried constraints. The model is instructed to identify and return the names of candidates who satisfy all the constraints based on the provided context. No exemplars are included.

Information Exclusion In certain scenarios, rather than the entire context, we may specifically require the model to focus on a subset of context and ignore the rest. One such scenario could be the survey responses, where we want to ignore the pilot responses from the authors of the survey. The other real-world example is suspicious data, where it is preferable to ignore those potential unreliable data to prevent them from contaminating the entire dataset. In the other words, this unit task examines the LLM's capability of helping with the data preprocessing step. In our setup, it is implemented upon the filtering task, where the model has to exclude certain candidates while completing the filtering task.

Chapter 4

Experiments

4.1 Models

We evaluate the performance of five state-of-the-art open and closed language models with various context length: Claude2 [3], Mistral-7B [17], Qwen1.5-7B [5], Gemini1 [14], and GPT4-Turbo[23]. Greedy decoding is used when generating outputs. We hope to generalize the correlation between performance on different long-context tasks across different LLMs to help with the prediction. See Table 4.1 for more details.

Model	Model Size	Context Token Limit	Openness
GPT4-Turbo	unknown	128k	closed
Claude2	unknown	100k	closed
Gemini1	unknown	32k	closed
Mistral	7B	32k	open
Qwen1.5	7B	32k	open

Table 4.1: Model cards for LLMs we employ, all featuring context lengths surpassing 32,000 tokens.

4.2 Evaluation Methods and Metrics

All our unit tasks are capable of automated evaluation, without reliance on other language models or human annotators. In the Key Retrieval task, we verify if the model's response includes the correct contextual segment and utilize a matching percentage as the evaluation criterion. In the Filter by Predicates and Information Exclusion task, we search for the presence of all candidate names and employ accuracy, the sum of true positives and true negatives over the total number of candidates, as the evaluation metric. 100 examples/queries are included in each experiment.

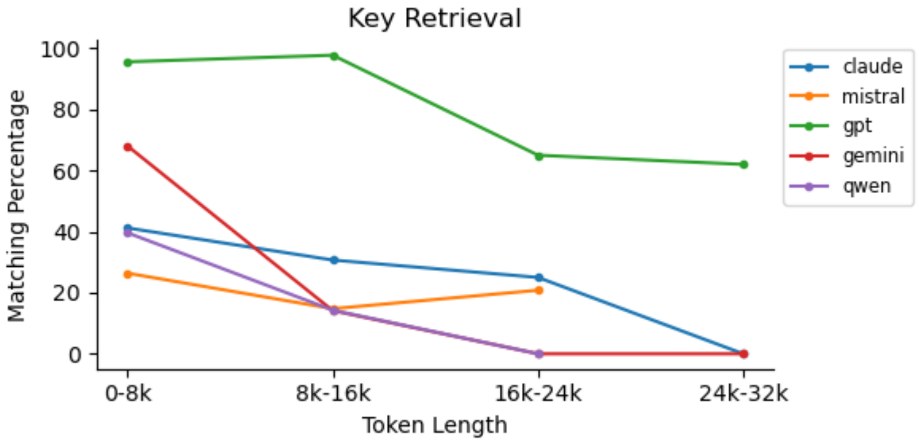

Figure 4.1: The experimental results of different models on the Key Retrieval task. The performance drops as the token length increases. The same set of thirty candidates is sampled for different models.

4.3 Results and Discussion

In this section, we present the performance of five language models across three unit tasks at varying token lengths. Overall, there's a noticeable decline in performance as token length extends. Furthermore, all unit tasks pose unexpected challenges to large language models, with performance falling considerably short of perfection.

Key Retrieval In Figure 4.1, we present the experimental results across different token length on the Key Retrieval unit task. We can see that there is a significant performance drop across all language models as the token length increases. resulting in several instances of performance dropping to zero with context between 16k and 24 tokens. Even the formidable GPT-4 struggles to maintain its performance beyond 16k tokens, achieving only around 60 percent matching.

Filter by Predicates Figure 4.2 contains the results of the Filter by Predicates unit task. Instead of achieving the optimal accuracy with shorter contexts, the models hit their best results with 16k-24k tokens. However, both Qwen and Claude experience a significant performance drop when the token length exceeds 24k.

Information Exclusion As shown in the Figure 4.3, adding an additional exclusion step affects model performance on the filtering task to various extents. The impact is most pronounced on contexts with median length (8k-24k). For extremely short contexts, models manage the additional exclusion step adeptly, while for extremely long contexts, disregarding two candidates doesn't significantly impair performance.

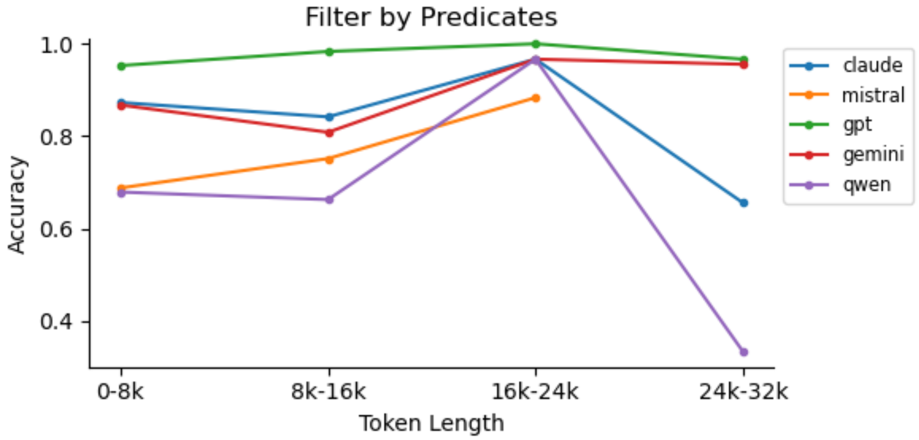

Figure 4.2: The experimental results of different models on the Filter by Predicates task. The models achieve their best performance with 16k-24k tokens. The same set of thirty candidates is sampled for different models.

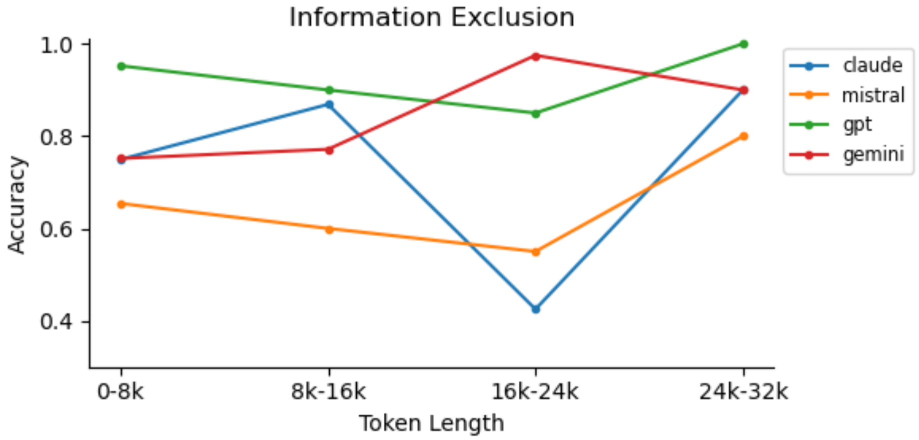

Figure 4.3: The experimental results on the Information Exclusion task. The most significant decline in performance occurs on contexts with median length. The same set of twenty candidates is sampled for different models, and the same two random candidates are instructed to be ignored.

Chapter 5

Analysis

5.1 Key Retrieval: Key Positions

We also explore the effect of varying the position of the key candidate within the context and observe its impact on performance. As illustrated in Figure 5.1, compared to a randomly positioned key, there is no notable alteration in retrieval quality across different key positions, even in scenarios involving extremely long contexts with 100 candidates. We hypothesize that language models are trained to mitigate the ”lost-in-the-middle” phenomenon [21], suggesting that they are better at utilizing relevant information situated at the beginning or end of their input context.

5.2 Filter by Predicates: Selectivity

Another exploratory experiment we undertake involves varying selectivity, which refers to the ratio of passing candidates to the total number of potential candidates provided in the context. We observe how this affects performance on the filtering task. Generally, we note a decrease in accuracy as selectivity increases from 0 percent to 30 percent (Figure 5.2). Higher selectivity presents greater challenges to language models as they must gather and return more information. The only exception is Gemini, whose performance reaches its peak at 20-30% selectivity.

5.3 Information Exclusion: Exclusion Proportion

We additionally explore whether adjusting the proportion of context instructed for exclusion impacts accuracy. We vary the exclusion proportion from 0 percent to 30 percent to simulate real-world scenarios. As shown in Figure 5.3, there is a consistent accuracy regardless of the exclusion proportion across all language models, suggesting the potential for feeding unprocessed information directly to language models.

5.4 Qualitative Analysis

Another interesting finding from our qualitative analysis of model outputs is that as the context length increases beyond certain thresholds where models struggle to manage effectively, instead of precisely addressing the initial parts of the context, the model tends to fail entirely, leading to inaccuracies even in information located at the beginning. Hence, it is crucial to determine the ”real” context limit of a specific language model for a particular task before throwing the entire context.

5.5 Synthetic Tasks over Numbers

Considering that language models face challenges with basic aggregation tasks in natural languages, how effectively can they perform such tasks—like MIN, MAX, and MODE—over numerical data? To explore this, we conducted experiments with discrete sets of numbers, examining how their performance varies with context length. We discovered that, similar to natural languages, performance is significantly influenced by context length, with most models struggling on basic aggregation tasks involving 500 or even 100 integers. See Appendix A.2 for details on the experimental setup and results.

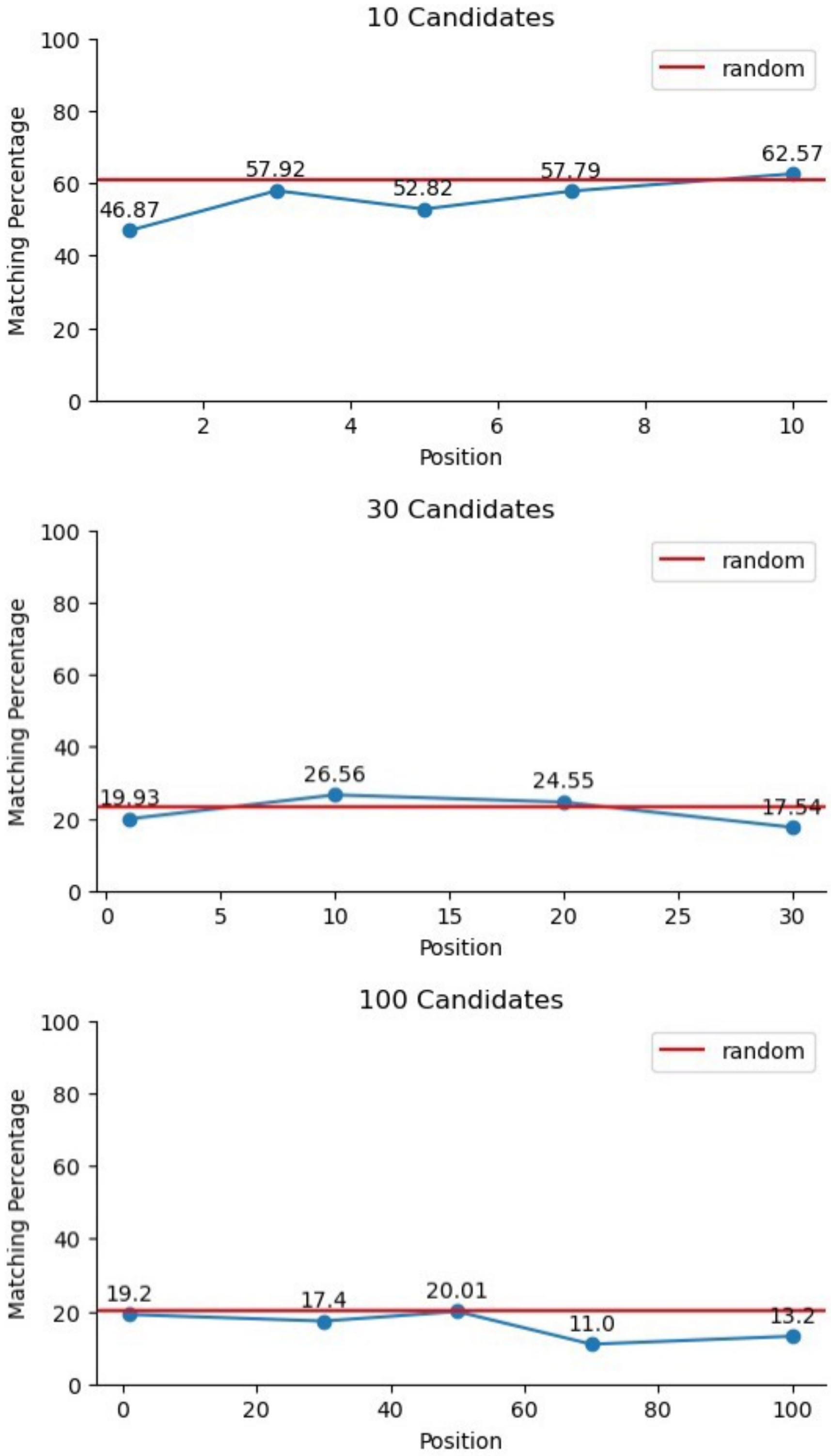

Figure 5.1: The impact of varying key candidate positions on retrieval quality across different context lengths using Claude2. Despite position variations, retrieval quality remains consistent, suggesting language models may mitigate the ”lost-in-the-middle” phenomenon.

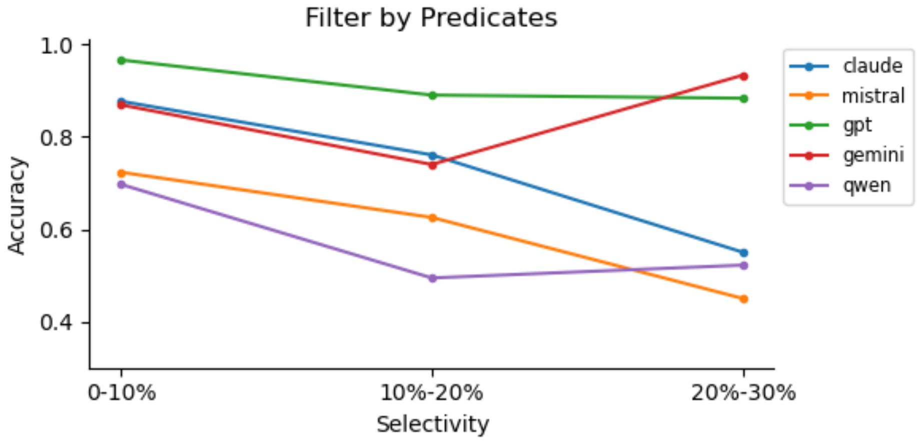

Figure 5.2: The impact of selectivity on filtering performance. As selectivity increases, accuracy generally declines, except Gemini.

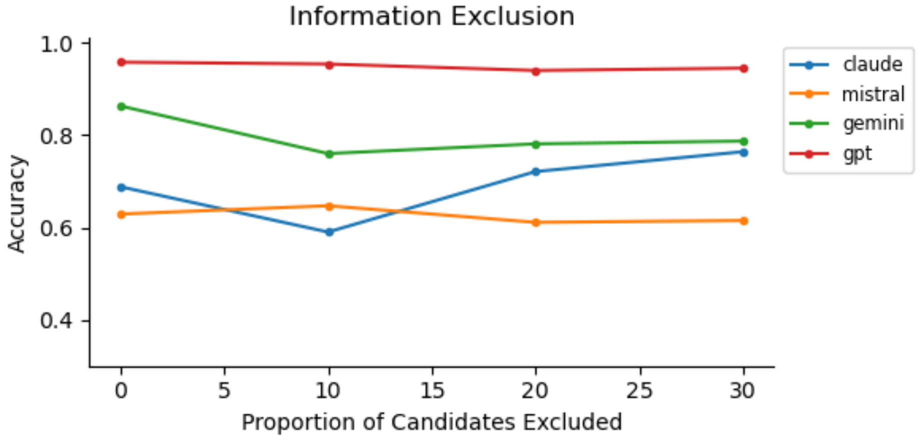

Figure 5.3: Experimental results of different exclusion proportion on accuracy across different language models. Results reveal consistent accuracy irrespective of the proportion excluded, suggesting the potential for direct input of unprocessed information.

Chapter 6

Towards Predicting Long-context Performance

In this section, we delve into the correlations across various tasks and different context lengths, aiming to offer insights into predicting long-context performance. Figure 6.1 illustrates the performance correlation between different unit tasks across varying token lengths, with the slopes indicated in the titles. It is evident that while different language models exhibit markedly different performances on the Key Retrieval task, their performance on the filtering task remains relatively consistent. Figure 6.2 and Figure 6.3 demonstrate the performance correlation between different token lengths under the same tasks. Models that perform exceptionally with short contexts typically also perform considerably well in the same task with longer contexts. By comparing the slopes, we hypothesize that, in terms of long-context prediction, the type of task provides more informative insights than context length; performance variations are more evident across different tasks than across different context lengths. Building upon this observation, we posit that assessing the long-context understanding of language models solely through the needle-in-the-haystack approach [18], despite its adaptable context length, lacks informativeness. Developing and evaluating additional unit tasks are imperative to obtain a more precise estimation of language models' performance on downstream long-context tasks.

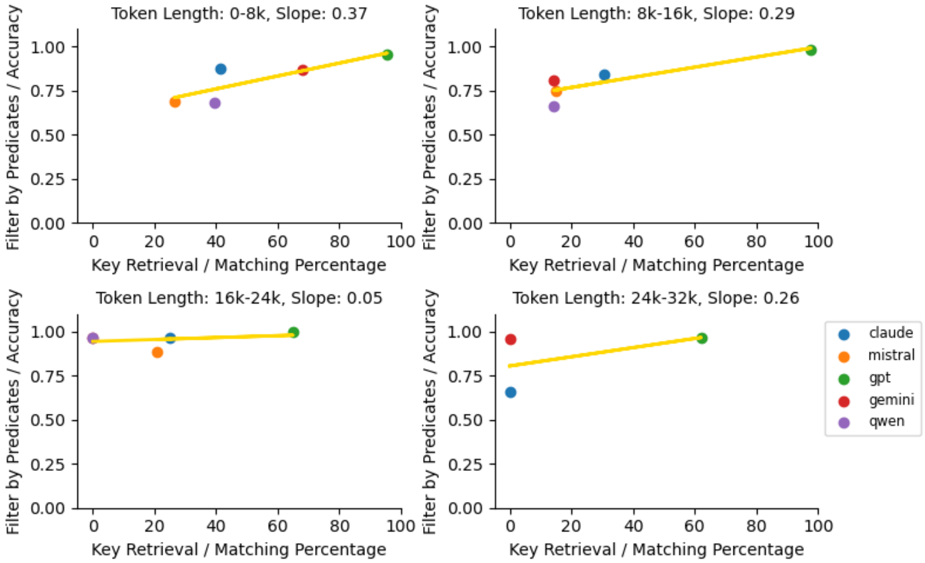

Figure 6.1: Correlation in performance across different tasks at varying token lengths, with a straight line fitted in each subplot. Despite significant performance discrepancies among different language models in the Key Retrieval task, their performance in the filtering task stays relatively consistent.

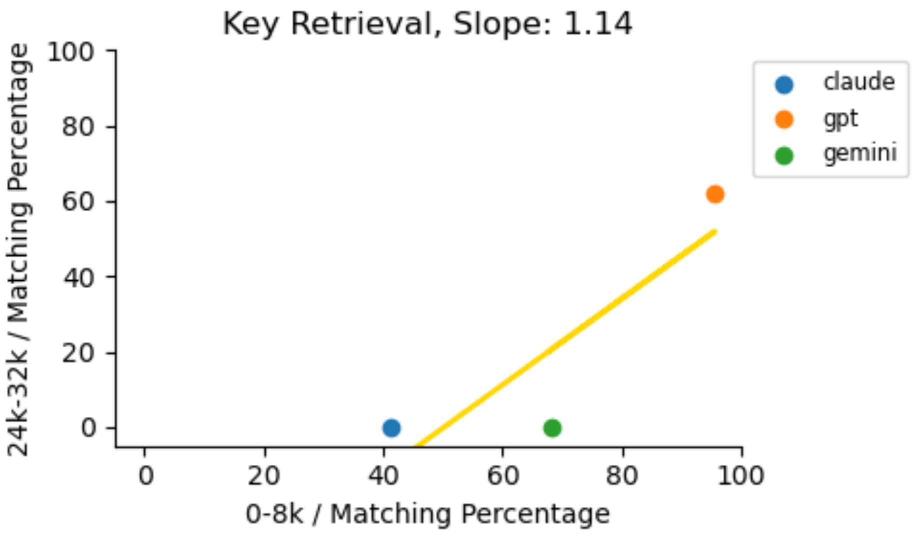

Figure 6.2: Performance correlation between 0-8k tokens and 24k-32k tokens in the Key Retrieval task, with a straight line fitted. Generally, models exhibiting better performance in the 0-8k token range also demonstrate better performance in the 24k-32k token range.

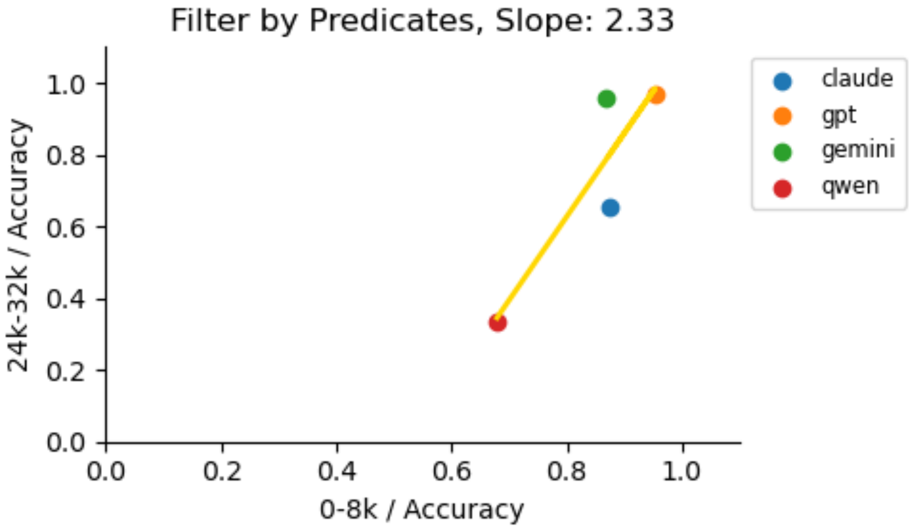

Figure 6.3: Performance correlation between 0-8k tokens and 24k-32k tokens in the Filter by Predicates task, with a straight line fitted. Generally, models exhibiting better performance in the 0-8k token range also demonstrate better performance in the 24k-32k token range.

Chapter 7

Conclusion and Future Work

In this study, we explore the potential factors associated with long-context performance by studying the correlation between different unit tasks and different context lengths. We propose a new multi-context evaluation dataset with three distinct tasks based on context accessing patterns, featuring adjustable context length, automated evaluation, and diagnosis beyond overall accuracy. All unit tasks present unexpected challenges for state-of-the-art large language models, and we observe a significant decrease in performance as token length increases. We hypothesize that, in terms of long-context prediction, the type of task provides more informative insights than context length and assessing the long-context understanding of language models solely through the needle-in-the-haystack approach [18] lacks informativeness.

While our work includes key retrieval, filtering, and exclusion unit tasks, future work could build upon this by incorporating more unit tasks like groupby to help with the prediction. The other thing we eagerly anticipate and find intriguing in the domain of long context understanding is the incorporation of subjective datasets, such as arguments or comments, as most current benchmarks draw from Wikidata and arXiv.

Bibliography

[1] Chenxin An et al. “L-eval: Instituting standardized evaluation for long context language models.” In: *arXiv preprint arXiv:2307.11088* (2023).

[2] Rohan Anil et al. “Palm 2 technical report.” In: *arXiv preprint arXiv:2305.10403* (2023).

[3] Anthropic. “Claude 2.” In: (2023).

[4] Anthropic. “The Claude 3 Model Family: Opus, Sonnet, Haiku.” In: (2024).

[5] Jinze Bai et al. “Qwen technical report.” In: *arXiv preprint arXiv:2309.16609* (2023).

[6] Yushi Bai et al. “Longbench: A bilingual, multitask benchmark for long context understanding.” In: *arXiv preprint arXiv:2308.14508* (2023).

[7] Tom Brown et al. “Language models are few-shot learners.” In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[8] Mirelle Bueno, Roberto Lotufo, and Rodrigo Nogueira. “Lissard: Long and Simple Sequential Reasoning Datasets.” In: *arXiv preprint arXiv:2402.07859* (2024).

[9] Shouyuan Chen et al. “Extending context window of large language models via positional interpolation.” In: *arXiv preprint arXiv:2306.15595* (2023).

[10] Zihang Dai et al. “Transformer-xl: Attentive language models beyond a fixed-length context.” In: *arXiv preprint arXiv:1901.02860* (2019).

[11] Jiayu Ding et al. “Longnet: Scaling transformers to 1,000,000,000 tokens.” In: *arXiv preprint arXiv:2307.02486* (2023).

[12] Zican Dong et al. “Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models.” In: *arXiv preprint arXiv:2309.13345* (2023).

[13] Google. “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.” In: *arXiv preprint arXiv:2403.05530* (2024).

[14] Google. “Gemini: A Family of Highly Capable Multimodal Models.” In: *arXiv preprint arXiv:2312.11805* (2024).

[15] Dirk Groeneveld et al. “Olmo: Accelerating the science of language models.” In: *arXiv preprint arXiv:2402.00838* (2024).

[16] Hongwei Han et al. “LUNA: language understanding with number augmentations on transformers via number plugins and pre-training.” In: *arXiv preprint arXiv:2212.02691* (2022).

[17] Albert Q Jiang et al. “Mistral 7B.” In: *arXiv preprint arXiv:2310.06825* (2023).

[18] G Kamradt. *Needle in a Haystack–pressure testing LLMs*. 2023.

[19] Jiaqi Li et al. “LooGLE: Can Long-Context Language Models Understand Long Contexts?” In: *arXiv preprint arXiv:2311.04939* (2023).

[20] Tianle Li et al. “Long-context LLMs Struggle with Long In-context Learning.” In: *arXiv preprint arXiv:2404.02060* (2024).

[21] Nelson F Liu et al. “Lost in the middle: How language models use long contexts.” In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173.

[22] Amirkeivan Mohtashami and Martin Jaggi. “Random-access infinite context length for transformers.” In: *Advances in Neural Information Processing Systems* 36 (2024).

[23] R OpenAI. “GPT-4 technical report.” In: *ArXiv* 2303 (2023).

[24] Bowen Peng et al. “Yarn: Efficient context window extension of large language models.” In: *arXiv preprint arXiv:2309.00071* (2023).

[25] Uri Shaham et al. “ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding.” In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 7977–7989.

[26] Hugo Touvron et al. “Llama: Open and efficient foundation language models.” In: *arXiv preprint arXiv:2302.13971* (2023).

[27] Tao Yuan et al. “LV-Eval: A Balanced Long-Context Benchmark with 5 Length Levels Up to 256K.” In: *arXiv preprint arXiv:2402.05136* (2024).

[28] Victor Zhong et al. “RoMQA: A Benchmark for Robust, Multi-evidence, Multi-answer Question Answering.” In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2023, pp. 7055–7067.

Appendix A

Appendix

A.1 Prompt Text

We present all adopted prompt texts in Table A.1 to facilitate reproducibility.

A.2 Synthetic Tasks over Numbers

This setting aims to assess the capability of language models to aggregate information over discrete sets of number and how this capability is impacted by the length of context.

Experimental Setup

In this set of experiments, we randomly generate 100 sets of integers ranging from 0 to 100 and ask for the MAX, MIN, MODE, and FILTER (count the number of integers less than a given threshold) in a zero-shot manner. Sets without unique mode are filtered out to ensure that there is a single, definitive answer for each task. We use accuracy as our evaluation metric. The four tasks are conducted given 10, 100, and 500 integers respectively to accommodate the input token limits.

We further introduce variations in the distribution of integers, driven by the hypothesis that these changes could introduce varying levels of difficulty for different tasks. For instance, when the majority of integers in a set cluster around the value of 20, and only a small subset is located near 80, it can become particularly challenging to identify the maximum value within that set.

This set of experiments analyzes three large language models, both closed and open, of varying sizes. These include Llama2-7B [26] and Llama2-13B [26], each with a maximum context length of 4096 tokens, and also Claude2 (100K) [3], using the Anthropic API.

Results and Discussion

Tables A.2, A.3, A.4, and A.5 presents the performance of LLMs across four aggregation tasks using discrete numbers generated from seven distinct distributions. The performance is compared with the corresponding baseline results, where a random integer is sampled from the input set as the output answer.

Overall, the length of context significantly impacts the performance of LLMs on aggregation tasks. As the length of context increase, their performance drop, while the performance on MAX and MIN may increase given 500 integers as the majority of maximum value is 100 and minimum value is 0 under the setting which make the tasks relatively easier.

Regarding the MAX and MIN tasks, the open-source model Llama2 demonstrates reasonably good performance when provided with a set of 10 integers, while it experiences a noticeable decline as the context length expands to 100 integers, and scaling up the model to the size of 13B parameters provides only marginal improvement in performance. The close-source model Claude, on the other hand, demonstrates remarkably strong performance, achieving an accuracy of over 0.9 when provided with a set of 500 integers across all distributions. Interestingly, MIN appears to present a greater challenge for LLMs compared to MAX, possibly due to the higher involvement of MAX in the pre-training data, and, overall, left-skewed distributions are easier for the MAX task while right-skewed distributions are easier for the MIN task, which meets the intuition and our hypothesize.

Based on the experimental results, MODE and FILTER tasks are more challenging for all LLMs. Specifically, LLMs struggle with the MODE task when provided with 500 integers, and encounter difficulty with the FILTER task when given 100 integers across all distributions. Their lack of confidence is evident as they tend to respond to the MODE task by providing a list of numbers rather than a single determined answer and a median number as an educated guess for the FILTER task, or generate code responses even with explicit instructions not to do so.

However, the poor results observed may be attributed to transformers' limited capacity to understand numerical information [16] rather than their deficient ability to aggregate information, and in the real world, these tasks can be perfectly solved by programs.

Task	Prompt
Key Retrieval	Find and return all passages related to ”bitcoin” from the following. Twitterrific: Twitterrific is a Mac OS X and iOS client for the social networking site Twitter. Mod4Win: It was one of the first Mod players for the Windows platform. Atom: Atom is a free and open-source text and source code editor for OS X, Linux, and Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub. Polyworld: Polyworld is a cross-platform (Linux, Mac OS X) program written by Larry Yaeger to evolve Artificial Intelligence through natural selection and evolutionary algorithms. MongoDB: MongoDB (from humongous) is a cross-platform document-oriented database. Stella: It is open source, and runs on most major modern platforms including Amiga, Windows, Mac OS X, Linux, Windows CE/Mobile, Dreamcast, GP2X, Nintendo DS, and Wii. BetterZip: It is developed solely for the OS X platform. Qt Jambi: Qt Jambi is a Java binding of the cross-platform application framework Qt. Overlayfs: OverlayFS is a filesystem service for Linux which implements a union mount for other file systems. bitcoin: In February 2014, a beta version of a Windows/Linux Firefox plug-in called FreeSpeechMe was released that allows automated resolution of .bit addresses, by downloading the Namecoin block chain and running it in the background. Answer: In February 2014, a beta version of a Windows/Linux Firefox plug-in called FreeSpeechMe was released that allows automated resolution of .bit addresses, by downloading the Namecoin block chain and running it in the background. Find and return all passages related to ”{query_candidate}” from the following. {candidates_with_context} Answer:

Filter by Predicates	List all passing candidates if any. (Names of candidates only)
	Query: {query}
	Candidates: {candidates_with_context}
	Answer:

Information Exclusion	List all passing candidates if any. (Names of candidates only) Ignore {ignore_candidates}.
	Query: {query}
	Candidates: {candidates_with_context}
	Answer:

Table A.1: Prompts of tasks introduced in this work.

Distribution	**Number of Integers**	**Llama2-7B**	**Llama2-13B**	**Claude2**
Random	10	0.78	0.96	**1.00**
	100	0.47	0.55	**0.97**
	500	0.59	0.62	**0.98**
Unimodal	10	0.84	**0.97**	0.91
	100	0.41	0.80	**0.97**
	500	0.43	0.55	**0.91**
Left-Skewed Unimodal	10	0.89	**0.96**	0.94
	100	0.55	**0.67**	0.43
	500	0.54	0.55	**0.97**
Right-Skewed Unimodal	10	0.82	**0.96**	0.93
	100	0.32	0.71	**0.99**
	500	0.20	0.35	**0.81**
Bimodal	10	0.80	**0.99**	0.98
	100	0.30	0.64	**1.00**
	500	0.57	0.72	**0.97**
Left-Skewed Bimodal	10	0.83	**0.92**	0.87
	100	0.74	0.75	**0.98**
	500	0.82	0.70	**0.95**
Right-Skewed Bimodal	10	0.81	0.98	**0.99**
	100	0.08	0.78	**1.00**
	500	0.41	0.66	**0.93**

Table A.2: Results for the MAX task. FAIL denotes performance below the baseline results and **bold** indicates best.

Distribution	**Number of Integers**	**Llama2-7B**	**Llama2-13B**	**Claude2**
Random	10	0.36	0.68	**0.83**
	100	0.08	0.20	**0.96**
	500	0.05	0.10	**0.96**
Unimodal	10	0.44	0.84	**0.85**
	100	0.03	0.25	**0.97**
	500	FAIL	0.08	**0.95**
Left-Skewed Unimodal	10	0.66	**0.97**	0.84
	100	0.08	0.29	**0.88**
	500	FAIL	0.03	**0.90**
Right-Skewed Unimodal	10	0.45	**0.83**	0.81
	100	0.17	0.22	**0.98**
	500	0.15	0.04	**0.93**
Bimodal	10	0.51	**0.87**	0.78
	100	0.16	0.42	**0.99**
	500	0.10	0.04	**0.97**
Left-Skewed Bimodal	10	0.33	0.69	**0.90**
	100	FAIL	0.10	**0.54**
	500	FAIL	0.04	**0.58**
Right-Skewed Bimodal	10	0.56	**0.87**	0.78
	100	0.12	0.27	**0.96**
	500	0.30	0.05	**0.99**

Table A.3: Results for the MIN task. FAIL denotes performance below the baseline results and **bold** indicates best.

Distribution	**Number of Integers**	**Llama2-7B**	**Llama2-13B**	**Claude2**
Random	10	0.76	0.92	**1.00**
	100	0.14	0.17	**0.23**
	500	0.08	**0.10**	0.04
Unimodal	10	0.88	0.86	**1.00**
	100	0.29	**0.30**	0.25
	500	**0.14**	0.08	0.05
Left-Skewed Unimodal	10	0.85	0.93	**1.00**
	100	**0.29**	0.26	0.03
	500	0.07	**0.15**	0.07
Right-Skewed Unimodal	10	0.74	0.87	**1.00**
	100	0.18	0.22	**0.34**
	500	**0.20**	0.04	0.19
Bimodal	10	0.60	0.91	**1.00**
	100	0.19	0.21	**0.34**
	500	0.07	0.02	**0.14**
Left-Skewed Bimodal	10	0.89	0.91	**1.00**
	100	0.32	**0.33**	0.28
	500	0.02	0.08	**0.17**
Right-Skewed Bimodal	10	0.81	0.94	**1.00**
	100	0.26	0.27	**0.49**
	500	**0.25**	0.03	0.15

Table A.4: Results for the MODE task. FAIL denotes performance below the baseline results and **bold** indicates best.

Distribution	**Number of Integers**	**Llama2-7B**	**Llama2-13B**	**Claude2**
Random	10	0.18	0.21	**0.22**
	100	FAIL	**0.04**	0.01
	500	FAIL	FAIL	FAIL
Unimodal	10	0.11	FAIL	**0.34**
	100	**0.03**	0.01	FAIL
	500	FAIL	FAIL	FAIL
Left-Skewed Unimodal	10	0.01	FAIL	**0.02**
	100	FAIL	FAIL	FAIL
	500	**0.04**	FAIL	FAIL
Right-Skewed Unimodal	10	0.14	0.03	**0.35**
	100	FAIL	FAIL	FAIL
	500	FAIL	FAIL	FAIL
Bimodal	10	**0.27**	0.02	**0.27**
	100	FAIL	FAIL	FAIL
	500	FAIL	FAIL	FAIL
Left-Skewed Bimodal	10	FAIL	FAIL	**0.11**
	100	**0.08**	FAIL	FAIL
	500	FAIL	FAIL	FAIL
Right-Skewed Bimodal	10	0.21	0.10	**0.29**
	100	FAIL	FAIL	FAIL
	500	FAIL	FAIL	FAIL

Table A.5: Results for the FILTER task. FAIL denotes performance below the baseline results and **bold** indicates best.