Marathon: A Race Through the Realm of Long Context with Large Language Models

Anonymous ACL submission

Abstract

With the advancement of large language models (LLMs) and the expansion of their context windows, existing long-context benchmarks fall short in effectively evaluating the models' comprehension and reasoning abilities in extended texts. Moreover, conventional benchmarks relying on F1 metrics often inaccurately score responses: they may undervalue correct answers that differ from the reference responses and overvalue incorrect ones that resemble the reference texts. In response to these limitations, we introduce **Marathon**, a novel evaluation benchmark that adopts a multiple-choice question format. It is specifically designed to overcome the constraints of previous benchmarks and provide a rapid, precise, and unbiased appraisal of the long-context comprehension skills of large language models. We conducted comprehensive evaluations on the Marathon benchmark with a range of state-of-the-art LLMs and assessed the effectiveness of various optimization strategies tailored for long-context generation. We anticipate that the Marathon benchmark and its associated leaderboard will enable a more precise and equitable evaluation of LLMs' capabilities in understanding and reasoning over extended contexts.

1 Introduction

In the rapidly evolving landscape of artificial intelligence technologies, the emergence of large language models (LLMs), as exemplified by ChatGPT (OpenAI, 2023b), showcases notable capabilities. The influence of these models extends beyond the well-established ChatGPT, gaining increasing prominence across diverse sectors. Existing LLMs are typically built upon Transformer architectures, which demand memory and computational resources that grow quadratically with sequence length. Consequently, Transformer language models have historically been trained with relatively modest predetermined context windows. For instance, LLaMA (Touvron et al., 2023a) employs a context size of 2048 tokens, while Llama2 (Touvron et al., 2023b) utilizes a context size of 4096 tokens. However, the pre-defined size imposes constraints on LLMs in various applications, such as summarizing extensive documents or addressing lengthy questions.

Significant research efforts have been devoted to extending the context length of LLMs. Due to the prohibitive expense of training LLMs with extended context lengths from scratch, the predominant studies have endeavored to enhance the capabilities of LLMs to comprehend long contexts through fine-tuning. These methods encompass extending the context window (Chen et al., 2023b), incorporating recurrent memory (Bulatov et al., 2024), employing sparse attention mechanisms (Xiao et al., 2023a), and augmenting with external memory (Wang et al., 2023). Concurrently, an increasing multitude of benchmarks have been introduced to assess the long-context understanding capabilities of LLMs. LongBench (Bai et al., 2023b) stands out as the first bilingual, multi-task benchmark specifically designed for the assessment of long-context understanding. This dataset continues to depend on the F1 score, which evaluates the responses of LLMs against a predefined set of possible answers. LooGLE (Li et al., 2023b) encompasses intricate long dependency tasks, including event timeline reordering, comprehension/reasoning, and computation. Nevertheless, the diverse nature of model-generated content introduces a challenge, as these predefined answers may not encompass all valid responses, thereby diminishing the precision of assessing model performance. There is a growing demand for high-quality benchmarks characterized by significantly longer text lengths and more challenging tasks, ensuring comprehensive evaluations.

In this study, we introduce a novel benchmark named **Marathon**, designed for long-context under-

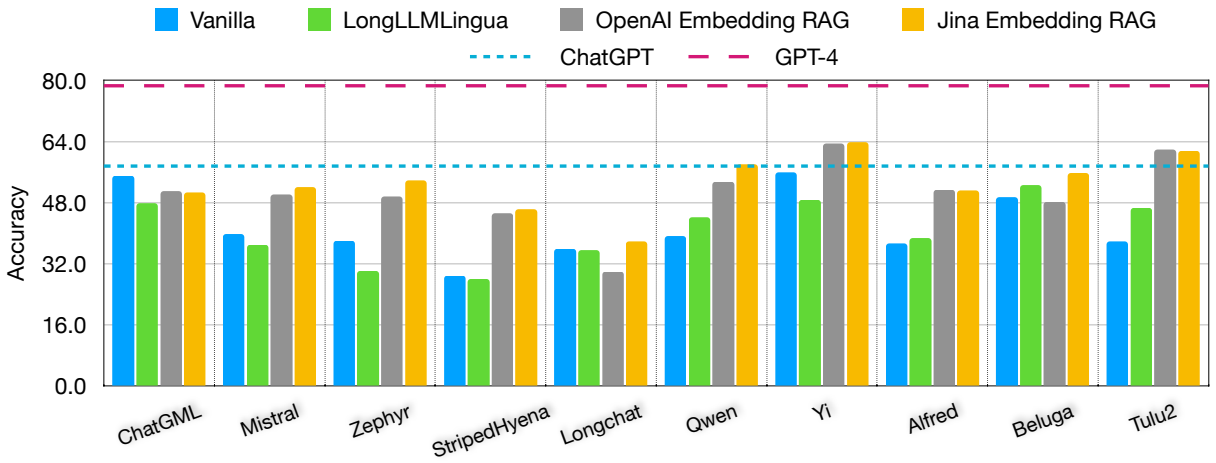

Figure 1: The overall accuracy of different models on Marathon. The x-axis represents the model, and the y-axis represents the average accuracy across all tasks. The different colors represent different methods of optimization.

standing and reasoning. In particular, this benchmark is constructed upon the foundations established by LooGLE (Li et al., 2023b) and LongBench (Bai et al., 2023b). The contextual lengths within this benchmark span from 2K to over 260K characters. For each extensive context provided, an associated question is paired with four meticulously crafted response options. These options have been carefully reviewed by humans and contain only one correct answer, with the remaining options designed to be highly misleading. This design makes the Marathon benchmark a particularly challenging one. The task for the large language model is to discern the accurate response option based on the extensive context provided.

The main contributions of this work are threefold:

- We introduce a novel multiple-choice long context benchmark that comprehensively evaluates the long context understanding and reasoning capabilities across 10 leading open-source large language models, as well as ChatGPT and GPT-4, covering six diverse types of tasks.
- We compare two prevalent methods for long context optimization (Prompt Compression and Retrieval Augmented Generation) along with two leading embedding models, assessing their impact on enhancing the long context reasoning abilities of large language models.
- Our findings reveal a general tendency among current open-source large language models to generate lengthier responses, accompanied by a notable deficiency in following instructions accurately.

2 Related Work

2.1 Prompt Compression

Although larger context windows enable large language models to handle longer contextual information, processing long-context information requires a significant amount of computing resources and places high demands on hardware. It also necessitates longer computational time, even in the inference stage. Therefore, some methods like LLMLingua (Jiang et al., 2023c) and LongLLMLingua (Jiang et al., 2023b) have been proposed to compress long contexts.

2.2 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) was originally proposed and applied to NLP tasks in (Lewis et al., 2020), and it has now become a mainstream method for improving the generation capability of large language models. RAG can extract the most relevant data from external knowledge bases and hand it over to the large language model for processing. This can alleviate the hallucination problem of large language models and enable people to trace the source of the content generated by large language models, ensuring the reliability of the generated content. Additionally, RAG can also be used to extract information from long documents that is most relevant to the user's query. This ensures that key information required to provide correct answers to questions is not lost while reducing the length of the context. Many projects such as

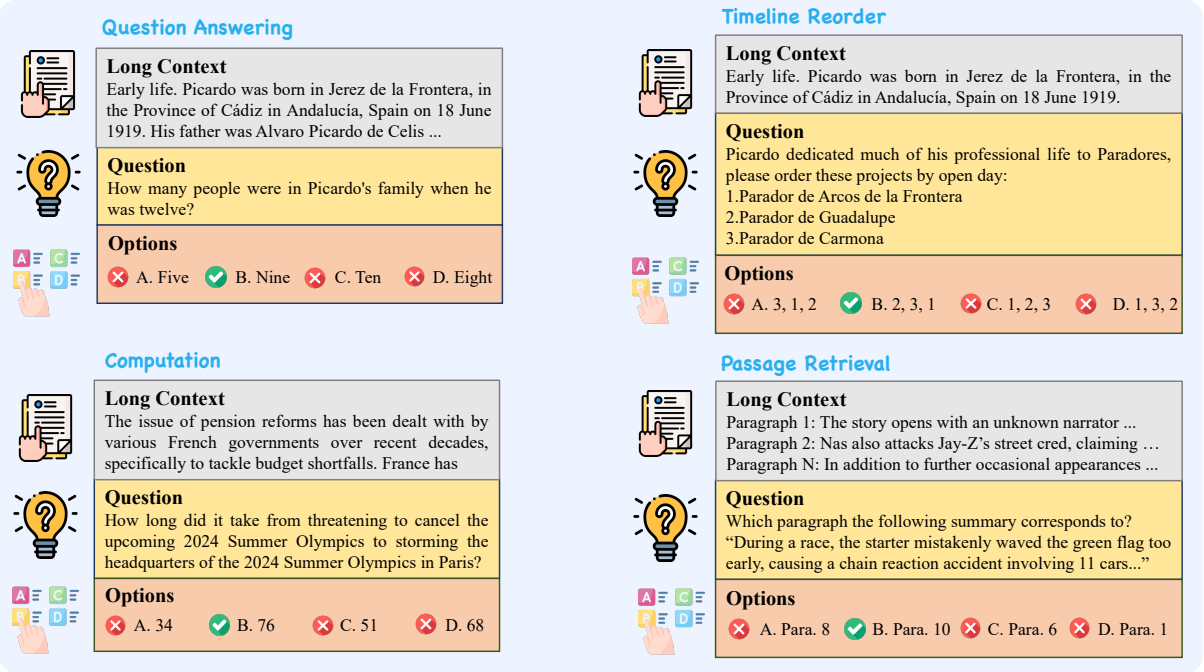

Figure 2: Examples of test case in benchmark, the context is truncated for display purposes.

longchain (longchain, 2022) and LlamaIndex (Liu, 2022) have achieved significant progress in combining RAG with large language models, greatly facilitating related research in this direction.

2.3 Long Context Models

The ability of large language models for handling long contexts has become increasingly important. ChatGPT (OpenAI, 2023a) supports a window size of 16k, while GPT-4 supports a window size of 128k, and Claude-2.1 supports a window size of 200k[1]. Many open-source large language models have started to expand the size of their context window. Longchat (Li et al., 2023a) and MPT (Team, 2023b) have achieved a window size of 16k, while Mistral (Jiang et al., 2023a) and Zephyr (Tunstall et al., 2023) have achieved a window size of 32k. By utilizing an adapted Rotary Embedding (Su et al., 2022) and sliding window (Beltagy et al., 2020) during fine-tuning, MistralLite, based on Mistral, has achieved a window size of 128k, enabling large language models to handle even longer contextual information.

2.4 Long Context Benchmarks

There have been many recent benchmarks used to assess the long context processing ability of large language models, such as LooGLE (Li et al., 2023b) and LongBench (Bai et al., 2023b). Liu et al. (2023b) on the other hand, noticed that the position of key information in long contexts greatly affects the capability of large language models to correctly understand and process text. Therefore, they used the NaturalQA (Kwiatkowski et al., 2019) dataset to construct a new benchmark to test the impact of different positions of key information in long context on the text processing capability of large language models.

Although LooGLE (Li et al., 2023b) and LongBench (Bai et al., 2023b) have constructed a relatively comprehensive set of evaluation tasks, the evaluation metrics used are still F1-score, Bleu or Rouge, which cannot accurately evaluate the ability of large language models to handle and understand long contexts.

3 Marathon

Present benchmarks for evaluating large language models primarily use a multiple-choice format, highlighted by studies such as MMLU (Hendrycks et al., 2021) and C-Eval (Huang et al., 2023). This multiple-choice approach helps prevent situations where large language models produce correct answers but are scored low due to missing corresponding elements in the reference answers, or when they produce incorrect answers that are scored high because they resemble the reference answers closely.

[1] https://www.anthropic.com/index/claude-2-1

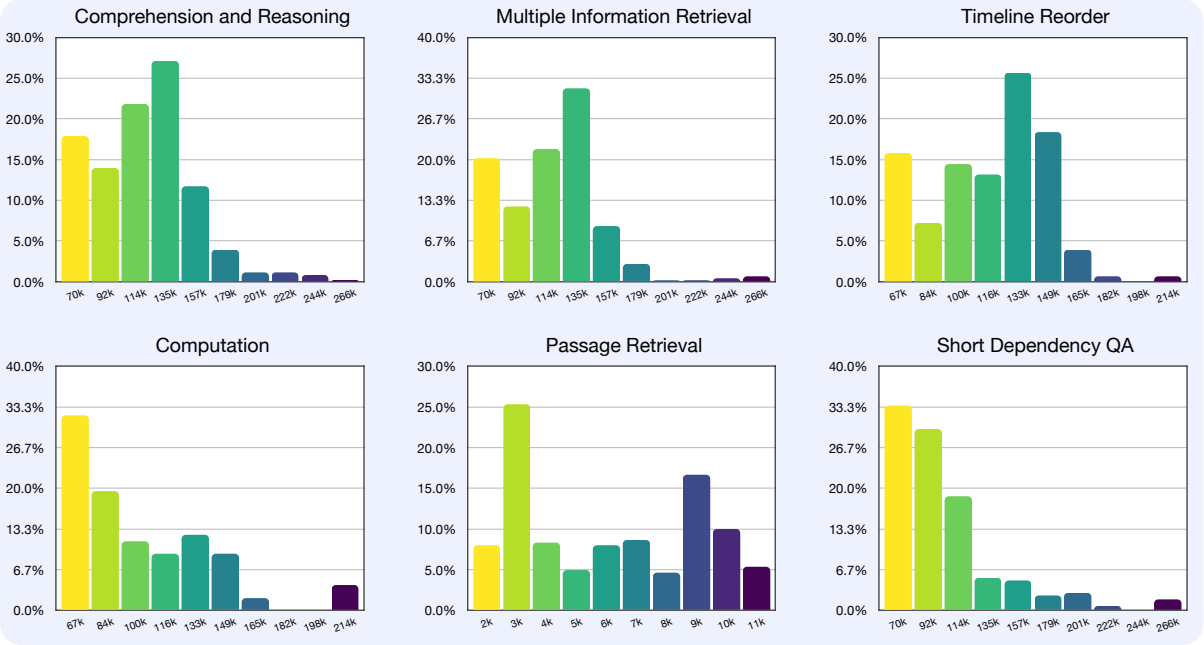

Figure 3: The distribution of context lengths for 6 tasks in the Marathon benchmark.

Therefore, influenced by LooGLE (Li et al., 2023b) and LongBench (Bai et al., 2023b), we developed a multiple-choice, long-context benchmark to more accurately evaluate the ability of large language models to understand extended contexts.

3.1 Overview

The Marathon benchmark includes six tasks: *Comprehension and Reasoning*, *Multiple Information Retrieval*, *Timeline Reorder*, *Computation*, *Passage Retrieval*, and *Short Dependency QA*. These tasks are grouped into four categories based on the type of questions they involve: Question Answering, Timeline Reorder, Computation, and Passage Retrieval. Table 1 provides the number of test samples for each task. Figure 2 presents example questions for each category.

3.2 Construction

All the test samples in the benchmark are in the form of multiple-choice questions, with each question containing one correct answer option and several distractor options. We use GPT-4 to generate the distractor options for each question. For each question, we divide the long context into multiple fragments of length 12,000 and randomly select one fragment. We require GPT-4 to generate three distractor options based on the given context fragment, question, and correct answer. The purpose of this approach is to avoid using excessively long context that exceeds GPT-4's context window, which may affect the accuracy of the generated results. By using shorter contexts, we can obtain distractor options that are more relevant to these shorter contexts.

Finally, to ensure the effectiveness and accuracy of these distractor options, we manually verify the options of each test sample.

3.3 Question Answering

Comprehension and Reasoning, *Multiple Information Retrieval* and *Short Dependency QA* are all types of traditional question-answer formats. The difference lies in the fact that *Comprehension and Reasoning*, *Multiple Information Retrieval* are selected from the Long Dependency QA dataset in LooGLE (Li et al., 2023b), while *Short Dependency QA* is selected from the Short Dependency QA dataset in LooGLE (Li et al., 2023b). In the question answering tasks, each question is accompanied by a corresponding long context, and the large language model is required to infer the correct answer according to the long context. For the *Short Dependency QA* task, the relevant content for the correct answer is relatively concentrated within the long context. For *Comprehension and Reasoning* and *Multiple Information Retrieval* tasks, the content relevant to the correct answer is more scattered throughout the long context. Therefore, the large language model needs to possess strong long context understanding capability in order to solve

the question correctly.

In the upper left of Figure 2, an example of a Question Answering task is provided. The question asks the large language model to answer a related question based on the content in the long context.

3.4 Timeline Reorder

Timeline Reorder task is a relatively novel question answering task. Unlike traditional question answering tasks, in the *Timeline Reorder* task, the question format requires large language models to sort a series of events described in a long context according to their chronological order. This task aims to examine the large language models' understanding of temporal relationships. Due to the dispersed distribution of events that need to be sorted by chronological order in the long context, large language models not only need to possess a correct understanding of temporal order but also require strong long context processing capabilities to answer correctly, which makes it a challenging task.

In the upper right of Figure 2, an example of the *Timeline Reorder* task is provided. The question requires the large language model to sort three events mentioned in the long context according to their chronological order.

3.5 Computation

Computation task is also different from traditional question answering tasks. Its question format involves providing a question related to numerical computation and requires the large language model to perform numerical calculations based on relevant content in the long context. For example, it may require calculating the number of children a certain character has at a specific time point, considering that the long context describes the character's life events, including the death of a child due to illness, which may affect the number of the character's offspring at subsequent time points. Therefore, to answer this question correctly, the large language model not only needs to be able to perform ordinary numerical calculations but also needs to capture all the key information related to the question. Compared to traditional computation and question answering tasks, this task is more challenging and can better reflect the large language model's capability to comprehend long context.

In the bottom left of Figure 2, an example of a *Computation* task is provided. The question requires the large language model to complete a numerical calculation question based on the content in the long context.

Task	No. Samples
Comprehension and Reasoning	357
Multiple Information Retreival	341
Timeline Reorder	152
Computation	97
Passage Retrieval	300
Short Dependency QA	283
Total	**1530**

Table 1: Statistics of Marathon.

3.6 Passage Retrieval

Passage Retrieval task is one form of task in the LongBench (Bai et al., 2023b). In order to enhance the diversity of our benchmark tasks, we have sampled 300 test data from the Passage Retrieval task in LongBench (Bai et al., 2023b), and reformed them into multiple-choice format using the method mentioned above. We have incorporated this task into our benchmark. The *Passage Retrieval* task requires large language models to locate the paragraph in a long context that corresponds to the given description in the question. Since the test data of the *Passage Retrieval* task is sampled from LongBench (Bai et al., 2023b), there are some limitations in terms of context length and timeliness. However, it remains a highly valuable task format. In future work, we will update its content to make it more suitable for the current needs of evaluating large language models.

A sample of the *Passage Retrieval* task is provided in the bottom right of Figure 2. The task requires large language models to locate the paragraph in a long context that corresponds to the given description in the question.

4 Experiments

4.1 Setup

Models. In this analysis, we incorporated a diverse array of models, distinguished by their parameter sizes, which span from 7B to 70B, and their context window capacities, extending from 8K to 200K. Additionally, the evaluation encompassed models constructed on a state-space architectural framework. The models scrutinized in this

investigation comprises ChatGLM3-6B-32K (Zeng et al., 2022; Du et al., 2022), Mistral-7B-Instruct-v0.1 (Jiang et al., 2023a), Zephyr-7B-β (Tunstall et al., 2023), StripedHyena-Nous-7B (Poli et al., 2023), Longchat-13B-16K (Li et al., 2023a), Qwen-14B-Chat (Bai et al., 2023a), Yi-34B (01.AI, 2023), Alfred-40B-1023 (Hallström et al., 2023), StableBeluga-2-70B (Mahan et al., 2023), Tulu-2-DPO-70B (Ivison et al., 2023), ChatGPT-1106 (OpenAI, 2023a), and GPT-4-1106-preview (OpenAI, 2023b).

Methods. In this evaluation, we first assessed the inherent ability of various models to comprehend long contexts. Then, we evaluated the current mainstream methods for handling long contexts: Compression and RAG. Specifically, for the compression method, we assessed LongLLMLingua (Jiang et al., 2023b), while for the RAG method, we evaluated two retrieval approaches, one based on OpenAI Embedding and the other on Jina Embedding (Günther et al., 2023).

4.2 Implementation Details

Prompt. We used the same prompt template to ask questions for all models, and required the answers to be returned in JSON format. The specific prompt format can be seen in Figure 5.

LongLLMLingua. For LongLLMLingua, we set the *compression rate* to 0.5, the *dynamic context compression ratio* to 0.4, We also sort the compressed contexts based on their importance.

Embedding RAG. For Embedding RAG, we utilize the ServiceContext and VectorStoreIndex of the Llama-Index (Liu, 2022). We employ various models as LLMs (Language Models), testing the OpenAI Embedding model and the Jina Embedding model as Embedding Models respectively. The default parameter settings are retained, with a chunk size of 1024 and a top-k value of 2. As for Jina Embedding, we set the pooling method to "mean" to align with Jina's encode implementation.

Hardware. All experiments in this evaluation were conducted on a server with 4*A100 80GB.

4.3 Results

4.3.1 Main results

The overall accuracy of various models on the Marathon benchmark is depicted in Figure 1. Detailed performance metrics of these models, utilizing distinct optimization techniques across a range of tasks, are presented in Table 2 within the appendix. To facilitate a more comprehensive comparative analysis of the outcomes, Figures 6, 7, 8, 9, 10, and 11 are provided in the appendix. The analysis indicates that the OpenAI Embedding Retrieval and Jina Embedding Retrieval models exhibit superior performance relative to the LongLLMLingua compression.

Moreover, all examined models exhibit diminished accuracy on both the *Timeline Reorder* and *Computation* tasks relative to their performance on alternative tasks. The implementation of the LongLLMLingua failed to yield any notable enhancements, and the advancements facilitated by the RAG were similarly constrained.

4.3.2 Vanilla

Within the subset of Vanilla method, the Yi-34B model, characterized by its 34 billion parameters, attains the highest accuracy, registering at 55.91%. This is closely followed by the ChatGLM3-6B-32K, which, despite its more modest parameter count of 6B, achieves an accuracy of 55.05%. Subsequently, the Beluga-70B model, notable for its context window limitation of 4K tokens, records an accuracy of 49.51%. The average accuracy observed across the remaining models does not exhibit significant variance, with none surpassing the 40% threshold.

4.3.3 LongLLMLingua

In contrast to the Vanilla approach, the implementation of LongLLMLingua yielded marginal improvements in accuracy for certain models: Qwen witnessed an enhancement of 4.85%, Alfred experienced a 1.51% increase, Beluga saw a 3.08% uplift, and Tulu2 benefited from an 8.64% augmentation. Conversely, this methodology had a detrimental effect on the performance of other models: ChatGLM3 encountered a 7.14% decrement in accuracy, Mistral suffered a 2.8% reduction, Zephyr experienced a significant 7.74% decrease, StripedHyena and Longchat showed a marginal decline of 0.10% and 0.26% respectively, and Yi's accuracy diminished by 7.25%.

4.3.4 OpenAI Embedding RAG

When juxtaposed with the baseline Vanilla methodology, the incorporation of OpenAI Embedding Retrieval notably enhances accuracy for several models: Mistral's accuracy improved by 10.37%, Zephyr's by 11.66%, StripedHyena's by 16.26%,

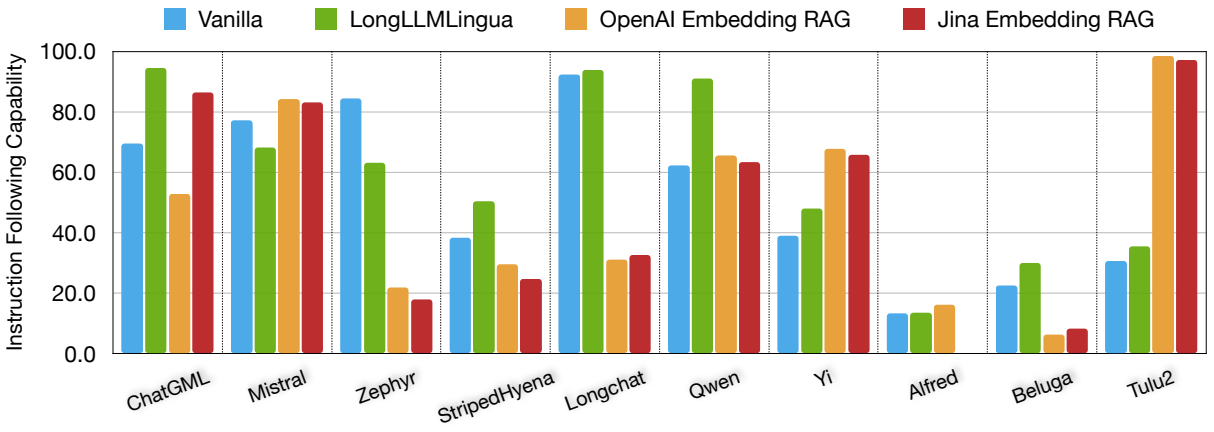

Figure 4: The instruction following capability of different models. The x-axis represents the model, and the y-axis represents the instruction following capability. The different colors represent different methods of optimization.

Qwen's by 14.19%, Yi's by 7.65%, Alfred's by 14.05%, and Tulu2's by an impressive 24.05%. Conversely, this approach has been observed to negatively impact the accuracy of certain models, with ChatGLM3 experiencing a 4.06% reduction, Longchat a 5.92% decrease, and Beluga a slight decline of 1.27%.

4.3.5 Jina Embedding RAG

Relative to the foundational Vanilla approach, the adoption of Jina Embedding Retrieval has led to accuracy enhancements across a majority of the evaluated models. Notably, Mistral's accuracy experienced a 12.23% increase, Zephyr's accuracy rose by 15.82%,StripedHyena's accuracy increased 17.37%, Longchat saw a 1.91% improvement, Qwen's accuracy was augmented by 18.15%, Yi's accuracy escalated by 7.9%, Alfred's accuracy advanced by 13.93%, Beluga's accuracy grew by 6.21%, and Tulu2's accuracy surged by 23.60%.

4.4 Instruction Following Capability

In our evaluation, numerous models exhibited limited ability to follow instructions accurately. We explicitly requested responses in JSON format, exemplified by a provided sample. Nonetheless, models occasionally responded in alternate formats or attempted JSON responses that were either incomplete or incorrect. Our statistical analysis, summarized in Table 3, categorizes responses as "JSON" for correct JSON format, "JSON-like" for flawed attempts at JSON due to errors like truncation or formatting issues, and "Plain Text" for responses in other formats. For a clearer comparison of models' ability to follow instructions after applying various optimizations, we focused on the rate of correct JSON responses as a measure of this capability, as depicted in Figure 4.

While Yi exhibited high accuracy in question answering, its compliance with instructions was notably lower, at 38.95%. In contrast, Beluga's adherence rate to instructions was even lower, at 22.48%, despite its capabilities. On the other hand, Longchat, despite its modest accuracy in answering questions, showcased a remarkable proficiency in following instructions, achieving a 92.29% compliance rate, closely trailing behind ChatGPT's 99.61%. The three distinct optimization techniques in our assessment demonstrated efficacy in diminishing the context length. However, it is noteworthy that none of these strategies consistently enhance the models' ability to follow instructions.

5 Discussion

Tendency of Long Responses. During our analysis, we observed that open-source large language models often generate lengthy responses, even with clear instructions for concise JSON-formatted answers. This tendency results in the generation of extraneous content, necessitating post-processing to isolate the needed information. Table 3 presents statistics on the models' output formats, highlighting their instruction-following capabilities. This issue likely stems from the models' training on predominantly long responses, making it challenging for them to comply with requests for brevity.

State Space Models. Recent studies, such as Mamba (Gu and Dao, 2023), highlight the advantages of state space models (SSMs) for long context reasoning tasks. StripedHyena (Poli et al., 2023) innovatively merges SSMs with transformer struc-

tures, indicating a new direction in large language models. Despite these advancements, our analysis reveals that StripedHyena underperforms in detailed long conext question answering compared to traditional transformers and does not reduce memory usage effectively, even with advanced attention mechanisms like Flash Attention 2 (Dao, 2023). These findings suggest the need for further optimization in State Space Models.

JSON Format. During the recent OpenAI Developer Day[2], significant advancements in the capabilities of GPT-4 (OpenAI, 2023b) were unveiled by OpenAI, notably the introduction of parallel function invocation and the specification of response formats in JSON. The parallel function invocation allows for the concurrent execution of multiple utility functions by large language models, thereby facilitating the efficient completion of complex user tasks. Moreover, the integration of JSON format for responses is instrumental in ensuring the seamless transmission of parameters and retrieval of results during function invocation, which is critical for the interoperability and functionality of AGI systems.

6 Future Work

Document as Context. Following the enhancements introduced at OpenAI Developer Day, GPT-4 (OpenAI, 2023b) has been equipped with a *Knowledge Retrieval* feature. This allows the model to utilize user-uploaded documents for answering queries, marking a significant development in Retrieval-Augmented Generation (RAG) applications. This trend suggests that future large language models will likely adopt similar functionalities, impacting the evaluation methodologies for long-context question answering. Instead of embedding lengthy contexts into prompts, future benchmarks should focus on the models' ability to extract and utilize information from user-provided documents to respond to queries. This approach necessitates a reevaluation of current benchmarks to align with these emerging capabilities.

Multi-modal Long Context. Models such as GPT4V (OpenAI, 2023c) and Gemini (Team, 2023a) have exhibited robust capabilities in facilitating interactions that span both visual and linguistic modalities. Likewise, open-source counterparts, including LLaVA (Liu et al., 2023a) and MiniGPT-4 (Zhu et al., 2023), have demonstrated commendable performance in assessments tailored to multimodal contexts. The utility of such models extends to various real-world applications that necessitate the processing of multimodal, extensive contexts, exemplified by the comprehensive analysis and synthesis of corporate annual reports. These applications demand not only the capacity of large language models to comprehend and infer within long textual contexts but also necessitate the integration of visual understanding abilities. Presently, the open-source community is lack of benchmarks specifically designed to evaluate the proficiency of models in handling extended, multimodal contexts. Therefore, establishing a comprehensive benchmark for multimodal, long-context capabilities is of significant importance.

Evolving Online Benchmarks. The rapid advancement of large language models calls for evolving evaluation methods and benchmarks. Traditional static benchmarks, often compromised over time by data leakage and integration into training, become less effective for accurate assessments. Moreover, the development of benchmarks by isolated teams or researchers is not only inefficient but also faces challenges in continuously updating evaluation data.

The solution lies in dynamic, Online Benchmarks, which would draw on the collective expertise and resources of the open-source community to ensure a constantly updated repository of new tasks and evaluation methods. This model aims to keep pace with the fast-evolving capabilities of language models, offering a more effective and scalable assessment framework.

7 Conclusion

In this paper, we compared 10 open-source large language models, including variations in their parameter sizes and context windows, along with OpenAI's ChatGPT and GPT-4. We assessed two prevalent optimization techniques such as LongLLMLingua, and RAG. The experimental results indicate that RAG-based optimization enhances the performance of large language models within long-context scenarios for QA-type tasks. However, the improvement is limited for tasks involving Timeline Reorder and Computation. Despite high accuracy in question-answering, these models show limited ability in following instructions.

[2] https://devday.openai.com

Limitations

Context Length Distribution. As depicted in Figure 3, the distribution of context lengths within the Marathon benchmark exhibits a lack of uniformity. The test instances corresponding to the tasks of *Comprehension and Reasoning*, *Multiple Information Retrieval*, *Computation*, *Short Dependency QA*, and *Timeline Reorder* predominantly feature context lengths that are concentrated at, or below, 130K characters. Conversely, test instances with context lengths surpassing 200K characters are notably scarce.

The test instances for the *Passage Retrieval* task derive from the LongBench (Bai et al., 2023b) dataset, which accounts for the markedly shorter context lengths in comparison to those associated with the remaining five tasks. This discrepancy underlies the superior performance metrics achieved by all models on the *Passage Retrieval* task. It is our intention to revise the test instances for *Passage Retrieval* to ensure consistency in context lengths with the other tasks. Furthermore, our ongoing efforts are directed towards augmenting the test instances for the remaining tasks, with the objective of achieving a uniform distribution of context lengths ranging from 60K to 260K characters across all tasks.

Evaluation. This paper presents a preliminary evaluation of optimization techniques for long contexts, which is not all-encompassing. In terms of optimization strategies, our evaluation of the Retrieval-Augmented Generation (RAG) method was limited to the employment of the OpenAI and Jina Embedding systems, exemplifying leading commercial and open-source embedding models, respectively. However, constraints related to time and financial resources precluded the examination of several advanced embedding systems, such as Voyage (Voyage.AI, 2023), Cohere (Cohere.Team, 2023), and BGE Embeddings (Xiao et al., 2023b). In the case of the Prompt Compression approach, aside from LongLLMLingua, there are other techniques like MemWalker (Chen et al., 2023a) that merit future exploration to fully assess the advantages and drawbacks of each embedding model and optimization method.

Moreover, in scenarios involving long context, while model accuracy and adherence are crucial, the speed of inference and memory demand are also vital factors to consider. This area features a variety of sophisticated optimization methods, including H2O (Zhang et al., 2023) and StreamingLLM (Xiao et al., 2023a). Subsequent research will focus on evaluating the performance of these inference optimization methods in scenarios with extensive textual content, with an emphasis on their speed of inference, memory consumption, QA precision, and instruction following capability.

Ethical Considerations

Data Source and Use. The benchmark leverages datasets that are publicly available and designated for research purposes. We have ensured that the use of these datasets adheres to their respective licenses and terms of use, emphasizing that our utilization is strictly confined to academic and research contexts.

Content Sensitivity and Bias. Our benchmark has been meticulously curated to exclude any content that could be deemed sensitive, such as violence, discriminatory language, or adult material.

Transparency and Reproducibility. In the spirit of fostering an open and fair research community, we will make the questions, contexts, and options of our benchmark's test cases publicly available. However, to maintain the integrity of the evaluation process, the correct answers to the test cases will not be disclosed. Instead, we will provide an online evaluation platform where researchers can submit their models' responses for assessment. This system is designed to ensure fairness and objectivity in the benchmarking process, allowing for an equitable comparison of different models' capabilities.

References

01.AI. 2023. Yi-series.

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

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023b. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail S. Burtsev. 2024. Scaling transformer to 1m tokens and beyond with rmt.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023a. Walking down the memory maze: Beyond context limit through interactive reading.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023b. Extending context window of large language models via positional interpolation.

Cohere.Team. 2023. Cohere embeddings. `https://txt.cohere.com/multilingual/`.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents.

Oskar Hallström, Amélie Chatelain, Clément Thiriet, Julien Séailles, Adrien Cavaillès, and Axel Marmet. 2023. Alfred-40b-1023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

Huiqiang Jiang, Qianhui Wu, , Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *ArXiv preprint*, abs/2310.06839.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023c. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023a. How long can open-source llms truly promise on context length?

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023b. Can long-context language models understand long contexts?

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Jerry Liu. 2022. LlamaIndex.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023b. Lost in the middle: How language models use long contexts. ArXiv:2307.03172.

longchain-ai longchain. 2022. LangChain.

Dakota Mahan, Ryan Carlow, Louis Castricato, Nathan Cooper, and Christian Laforte. 2023. Stable beluga models.

OpenAI. 2023a. Chatgpt. https://chat.openai.com.

OpenAI. 2023b. Gpt-4 technical report.

OpenAI. 2023c. Gpt4v system card. https://openai.com/research/gpt-4v-system-card.

Michael Poli, Jue Wang, Stefano Massaroli, Jeffrey Quesnelle, Ryan Carlow, Eric Nguyen, and Armin Thomas. 2023. StripedHyena: Moving Beyond Transformers with Hybrid Signal Processing Models.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2022. Roformer: Enhanced transformer with rotary position embedding.

Gemini Team. 2023a. Gemini: A family of highly capable multimodal models.

MosaicML NLP Team. 2023b. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

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

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Voyage.AI. 2023. Voyage embeddings. https://docs.voyageai.com/embeddings/.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. Augmenting language models with long-term memory.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023a. Efficient streaming language models with attention sinks.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023b. C-pack: Packaged resources to advance general chinese embedding.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2o: Heavy-hitter oracle for efficient generative inference of large language models.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models.

A Template and Prompt

Figure 5 illustrates the prompt used in our model evaluation process. The system prompt is denoted in green, the provided long context in cyan, the question related to the long context in yellow, the quartet of options in blue, and the orange segment delineates the response format for the model, accompanied by an concrete example. The instructions and responses templates may vary across different models. To ensure consistency, we adjust the prompts during our evaluations to match the templates used during the models' training phases.

You are an expert at reading and analyzing lengthy texts for examinations. Your task is to carefully read the provided text, understand its content and details, and accurately answer multiple-choice questions about the text. Keep in mind that the correct answer must be based entirely on the content of the text, without including any external information or personal opinions.

Context:
Olympics on the Whistler Sliding Centre in Whistler, British Columbia, Canada. Hours later, the International Luge Federation concluded that the accident was caused by a steering error and not a track error; nevertheless, ...

Question:
Based on the description above, what is the name of son of lord krishna?

Options:
A. Jon Owen
B. Nodar Kumaritashvili
C. Ulrich Hahn
D. Paul Aste

Please answer this question with JSON format, for example {"option":"A"}.
Answer:

Figure 5: An example of test prompt, the context is truncated for display purposes.

B Detailed Evaluation Results

Table 2 presents the detailed performance metrics of various models, utilizing distinct optimization techniques across a range of tasks. C&R refers to *Comprehension and Reasoning* task; MIR refers to *Multiple Information Retrieval* task; TR refers to *Timeline Reorder* task; Com. refers to *Computation* task; PR refers to *Passage Retrieval* task; SDQA refers to *Short Dependency Question Answering* task; Avg. denotes the average accuracy across all tasks. To provide a more intuitive comparison of the effects of different optimization approaches on the long-context comprehension and reasoning capabilities of various models across different tasks, we also illustrated Figures 6, 7, 8, 9, 10, and 11.

C Detailed Instruction Following Capability

As shown in Figure 5, we asked the models to produce results in JSON format to assess how well they follow instructions based on their output format. Table 3 summarizes the performance of 10 open-source large language models in this regard. "JSON" means the output was exactly in JSON format. "JSON-like" refers to outputs that tried to be in JSON format but included mistakes or extra text. "Plain Text" covers outputs in other formats. Since ChatGPT and GPT-4 can always gave results in JSON format, they're not included in Table 3.

Model	Para.	CW	C&R	MIR	TR	Com.	PR	SDQA	Avg.
GPT-4	-	128K	77.03%	69.21%	69.08%	60.82%	100.00%	95.41%	78.59%
ChatGPT	-	16K	62.18%	51.32%	19.74%	34.02%	95.67%	81.27%	57.37%
Vanilla									
Chatglm	6B	32K	55.46%	46.63%	30.26%	37.11%	81.33%	79.51%	55.05%
Mistral	7B	32K	46.22%	41.94%	28.95%	23.71%	49.67%	48.41%	39.81%
Zephyr	7B	32K	41.46%	37.83%	33.24%	21.65%	47.67%	47.00%	37.97%
StripedHyena	7B	18K	30.25%	29.91%	25.00%	21.65%	24.00%	43.11%	28.99%
Longchat	13B	16K	37.25%	34.60%	28.29%	27.84%	42.00%	45.23%	35.87%
Qwen	14B	8K	45.38%	39.00%	26.32%	23.71%	56.33%	44.88%	39.27%
Yi	34B	200K	59.66%	47.21%	37.50%	36.08%	90.00%	65.02%	55.91%
Alfred	40B	8K	40.90%	39.30%	26.32%	20.62%	49.00%	47.70%	37.31%
Beluga	70B	4K	55.74%	43.70%	36.84%	36.08%	65.33%	59.36%	49.51%
Tulu2	70B	8K	46.50%	35.48%	30.26%	22.68%	46.33%	46.29%	37.92%
LongLLMLingua Compression									
Chatglm	6B	32K	47.06%	37.54%	25.66%	22.68%	98.33%	56.18%	47.91%
Mistral	7B	32K	40.06%	31.38%	23.03%	27.84%	57.00%	42.76%	37.01%
Zephyr	7B	32K	30.81%	26.39%	23.68%	18.56%	54.00%	27.92%	30.23%
StripedHyena	7B	18K	22.97%	20.23%	10.53%	15.46%	58.00%	41.34%	28.09%
Longchat	13B	16K	37.82%	28.74%	26.32%	20.62%	61.67%	38.52%	35.61%
Qwen	14B	8K	42.58%	36.66%	27.63%	26.80%	88.67%	42.40%	44.12%
Yi	34B	200K	49.58%	42.23%	30.26%	22.68%	90.33%	56.89%	48.66%
Alfred	40B	8K	38.94%	32.84%	26.32%	29.90%	59.00%	45.94%	38.82%
Beluga	70B	4K	50.42%	42.82%	36.84%	27.84%	94.00%	63.60%	52.59%
Tulu2	70B	8K	45.94%	35.19%	34.87%	12.37%	98.00%	53.00%	46.56%
OpenAI Embedding RAG									
Chatglm3	6B	32K	56.58%	43.40%	28.95%	28.87%	81.33%	66.78%	50.99%
Mistral	7B	32K	51.54%	47.21%	27.63%	27.84%	79.00%	67.84%	50.18%
Zephyr	7B	32K	52.38%	43.99%	28.29%	24.74%	76.67%	71.73%	49.63%
StripedHyena	7B	18K	48.46%	40.18%	32.24%	25.77%	62.67%	62.19%	45.25%
Longchat	13B	16K	38.10%	25.81%	19.08%	12.37%	43.00%	41.34%	29.95%
Qwen	14B	8K	61.34%	46.33%	31.58%	18.56%	93.00%	69.96%	53.46%
Yi	34B	200K	66.39%	55.13%	38.82%	42.27%	95.00%	83.75%	63.56%
Alfred	40B	8K	52.38%	48.39%	25.00%	27.84%	87.33%	67.14%	51.35%
Beluga	70B	4K	61.90%	46.33%	3.28%	21.65%	81.00%	75.27%	48.24%
Tulu2	70B	8K	64.99%	53.37%	41.45%	34.02%	95.67%	82.33%	61.97%
Jina Embedding RAG									
Chatglm	6B	32K	52.94%	44.57%	27.63%	23.71%	83.33%	71.38%	50.60%
Mistral	7B	32K	54.90%	43.99%	32.24%	25.75%	79.00%	76.33%	52.04%
Zephyr	7B	32K	52.66%	46.33%	30.92%	23.71%	91.00%	78.09%	53.79%
StripedHyena	7B	18K	45.10%	42.22%	30.92%	30.93%	64.67%	64.31%	46.36%
Longchat	13B	16K	42.58%	33.43%	22.37%	13.40%	57.67%	57.24%	37.78%
Qwen	14B	8K	60.50%	46.63%	44.08%	24.74%	94.33%	78.45%	58.12%
Yi	34B	200K	66.67%	54.25%	45.39%	38.14%	95.00%	83.39%	63.81%
Alfred	40B	8K	50.42%	44.28%	27.63%	25.77%	88.33%	71.02%	51.24%
Beluga	70B	4K	59.94%	49.85%	23.68%	27.84%	96.00%	77.03%	55.72%
Tulu2	70B	8K	64.99%	54.25%	38.82%	31.96%	95.00%	84.10%	61.52%

Table 2: The evaluation results of models on Marathon benchmark.

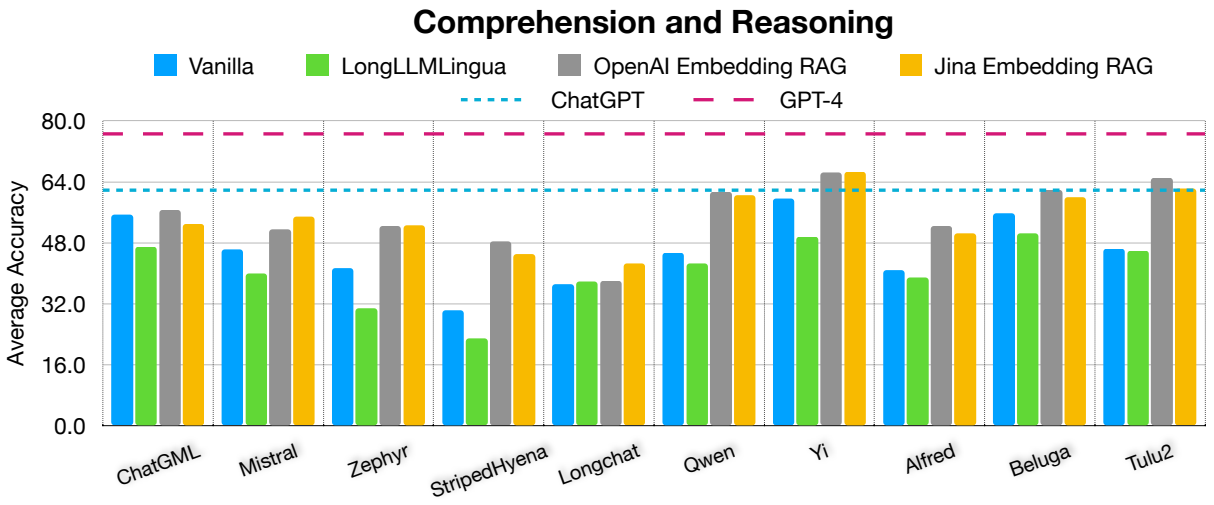

Figure 6: The performance of models on comprehension and reasoning task.

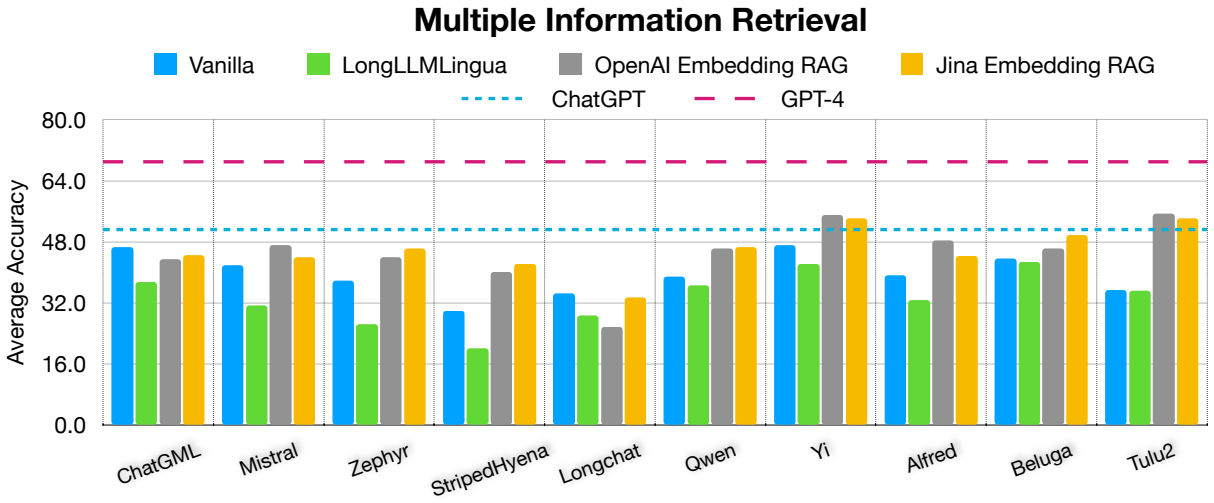

Figure 7: The performance of models on multiple information retrieval task.

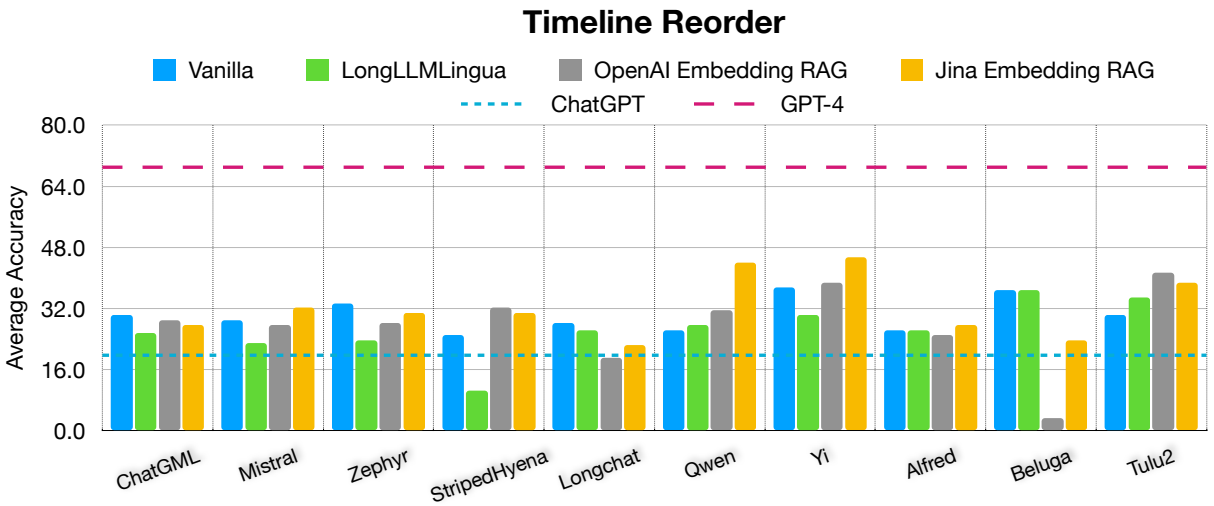

Figure 8: The performance of models on timeline reorder task.

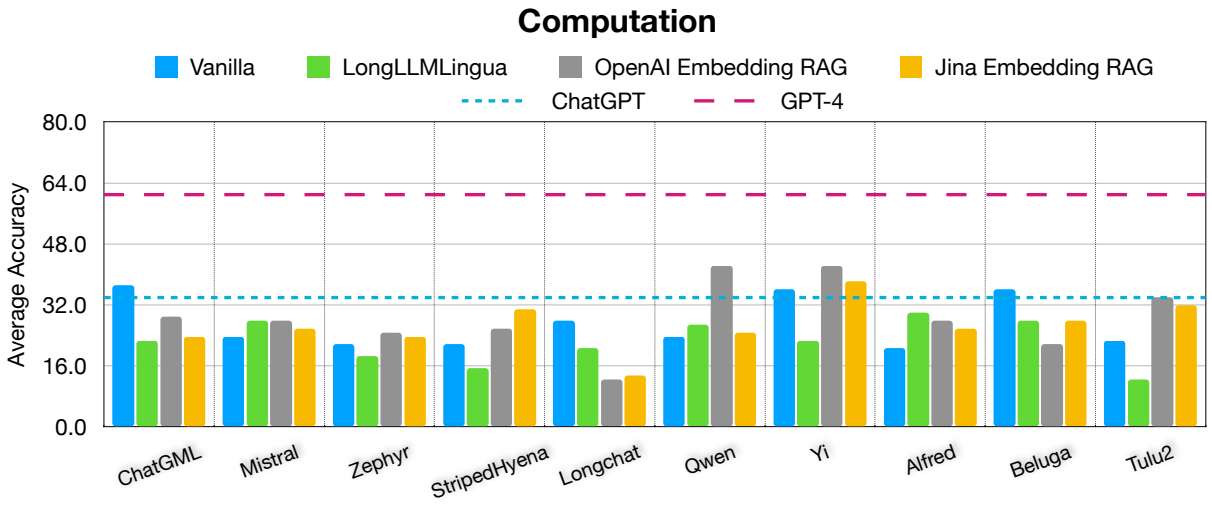

Figure 9: The performance of models on computation task.

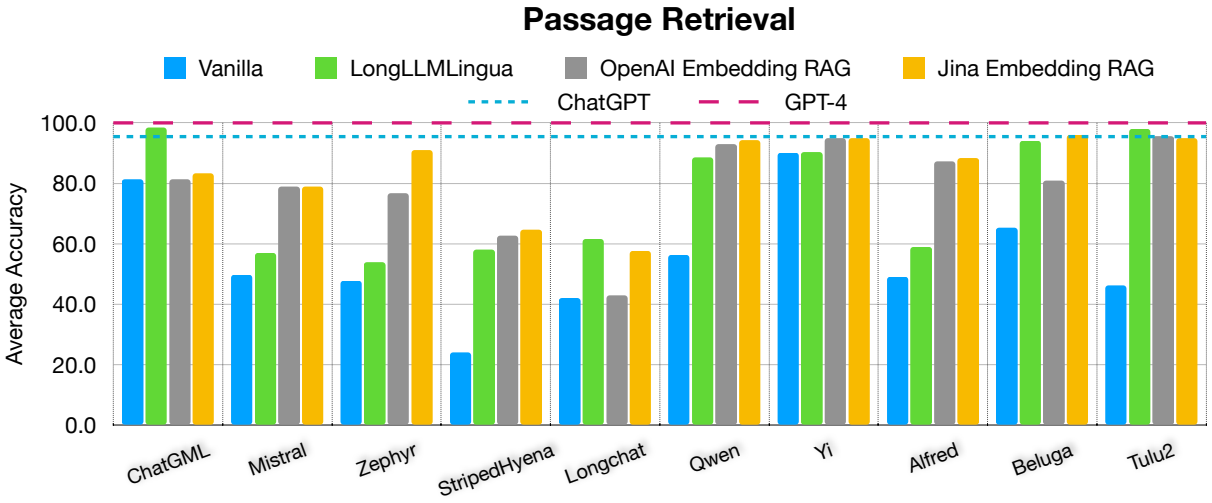

Figure 10: The performance of models on passage retrieval task.

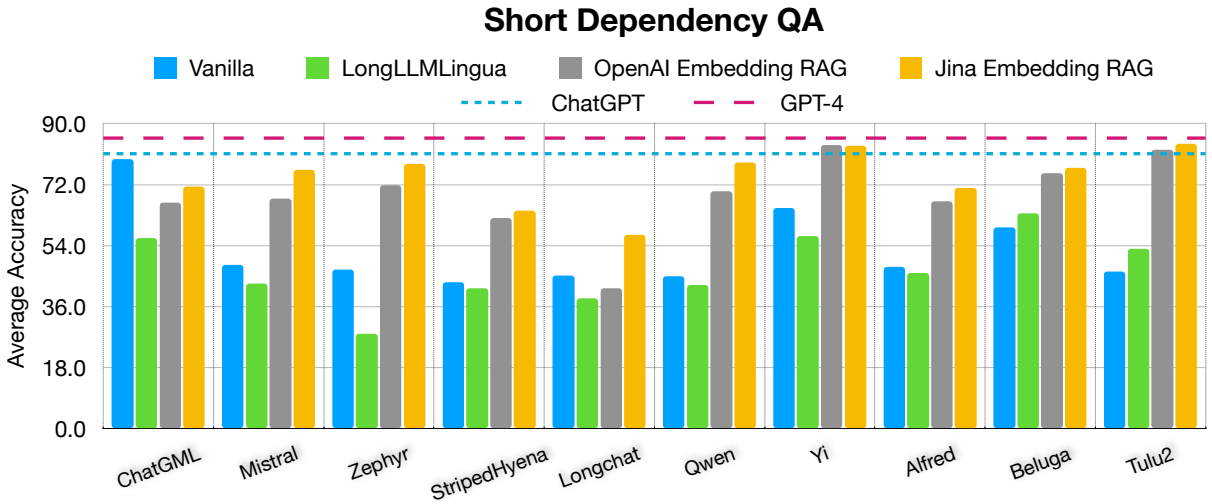

Figure 11: The performance of models on short dependency question answering task.

Type	Chatglm	Mistral	Zephyr	StripedHyena	Longchat	Qwen	Yi	Alfred	Beluga	Tulu2
Vanilla										
JSON	69.48%	77.22%	84.51%	38.43%	92.29%	62.29%	38.95%	13.27%	22.48%	30.72%
JSON-like	30.52%	21.18%	6.86%	28.95%	3.99%	0.72%	28.56%	81.96%	0.33%	46.47%
Plain Text	0.00%	6.60%	8.63%	32.61%	3.73%	36.99%	32.48%	4.77%	71.19%	22.81%
LongLLMLingua Compression										
JSON	94.58%	68.10%	63.20%	50.39%	93.92%	90.92%	48.10%	13.66%	29.97%	35.45%
JSON-like	5.42%	22.68%	9.15%	12.94%	2.81%	0.06%	26.60%	85.95%	0.59%	43.46%
Plain Text	0.00%	9.22%	27.65%	36.67%	32.68%	9.02%	25.29%	0.39%	69.54%	19.08%
OpenAI Embedding RAG										
JSON	52.88%	84.31%	21.83%	29.54%	31.11%	65.62%	67.71%	16.27%	6.27%	98.43%
JSON-like	42.42%	6.67%	69.87%	52.09%	23.73%	16.93%	32.16%	83.73%	0.00%	1.11%
Plain Text	4.71%	9.02%	8.30%	18.37%	45.16%	17.45%	0.13%	0.00%	93.73%	0.46%
Jina Embedding RAG										
JSON	86.34%	83.14%	17.91%	24.77%	32.75%	63.33%	65.69%	0.00%	8.43%	97.19%
JSON-like	9.15%	6.21%	73.86%	56.80%	21.70%	17.91%	34.18%	100.00%	0.007%	2.16%
Plain Text	4.51%	10.65%	8.24%	18.43%	45.56%	18.76%	0.13%	0.00%	91.50%	0.65%

Table 3: The evaluation results of large language models on the Marathon benchmark for instruction following.