More Documents, Same Length: Isolating the Challenge of Multiple Documents in RAG

Shahar Levy* **Nir Mazor*** **Lihi Shalmon***

Michael Hassid **Gabriel Stanovsky**

School of Computer Science and Engineering
The Hebrew University of Jerusalem, Jerusalem, Israel
{shahar.levy2, nir.mazor, lihi.shalmon, michael.hassid, gabriel.stanovsky}@mail.huji.ac.il

Abstract

Retrieval-Augmented Generation (RAG) enhances the accuracy of Large Language Model (LLM) responses by leveraging relevant external documents during generation. Although previous studies noted that retrieving many documents can degrade performance, they did not isolate how the quantity of documents affects performance while controlling for context length. We evaluate various language models on custom datasets derived from a multi-hop QA task. We keep the context length and position of relevant information constant while varying the number of documents, and find that increasing the document count in RAG settings poses significant challenges for most LLMs, reducing performance by up to 20%. However, Qwen2.5 maintained consistent results across increasing document counts, indicating better multi-document handling capability. Finally, our results indicate that processing multiple documents is a separate challenge from handling long contexts. We also make the datasets and code available[1] to facilitate further research in multi-document retrieval.

1 Introduction

The RAG approach enriches prompts with relevant documents, retrieved according to an input query (Karpukhin et al., 2020). For example, given a question about a certain historical period, RAG techniques can retrieve documents related to the time from a large historical corpus.

Recent work has noted a drop in RAG performance when retrieving many documents. For example, in multi-hop QA, LLMs struggle when the number of retrieved documents grows, even when presented with all the needed information (Press et al., 2022; Liu et al., 2023; Levy et al., 2024; Wang et al., 2024). Such deficiencies were observed without controlling for the *number of tokens* in which the information is conveyed, i.e., when the number of documents grew, so did the number of overall tokens, thus conflating between the challenge of long context and multi document.

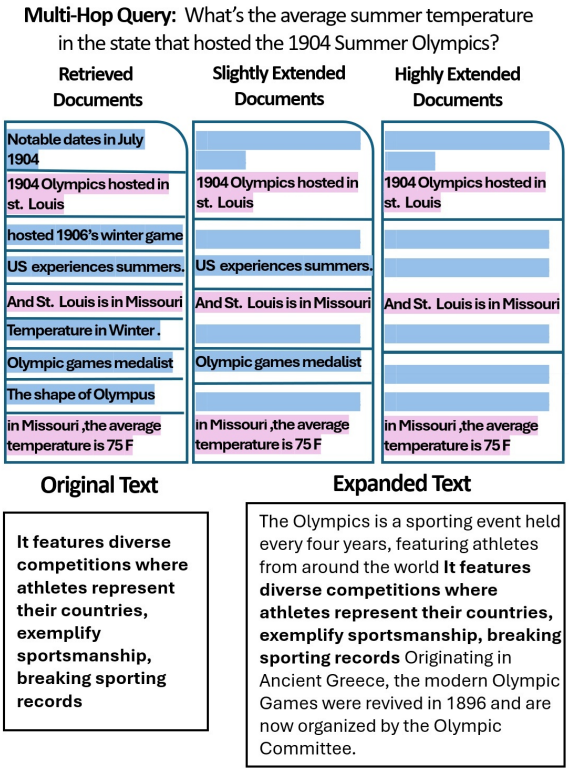


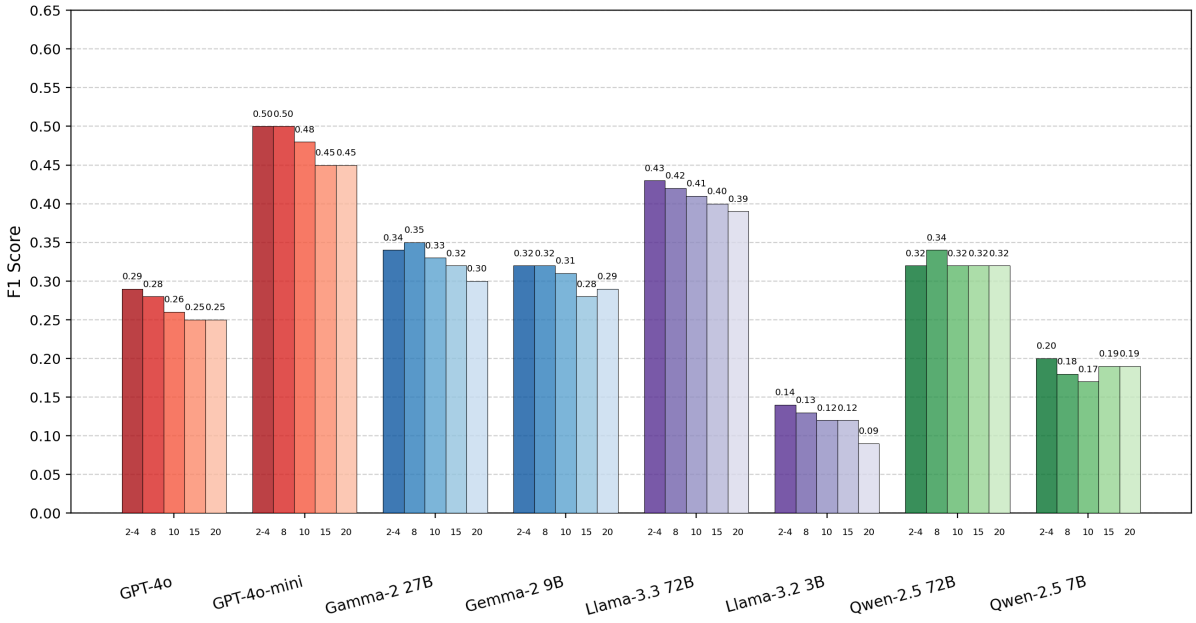
Figure 1: **More Documents, Same Length.** We create various sets containing the same questions but differing in the number of distractor documents. Each set includes a multi-hop question, all of the supporting documents that contain the information to answer the question (pink), and varying distractor documents (blue).We begin with either 10 or 20 documents (depending on the dataset) as our full-doc version (left) and then reduce the number of documents while maintaining a fixed context size. When fewer documents are used, the remaining documents are extended (blue without text) so that concatenating them yields the same total length.

In this work, we address the following question: *Assuming a fixed input length, how is LLM per-*

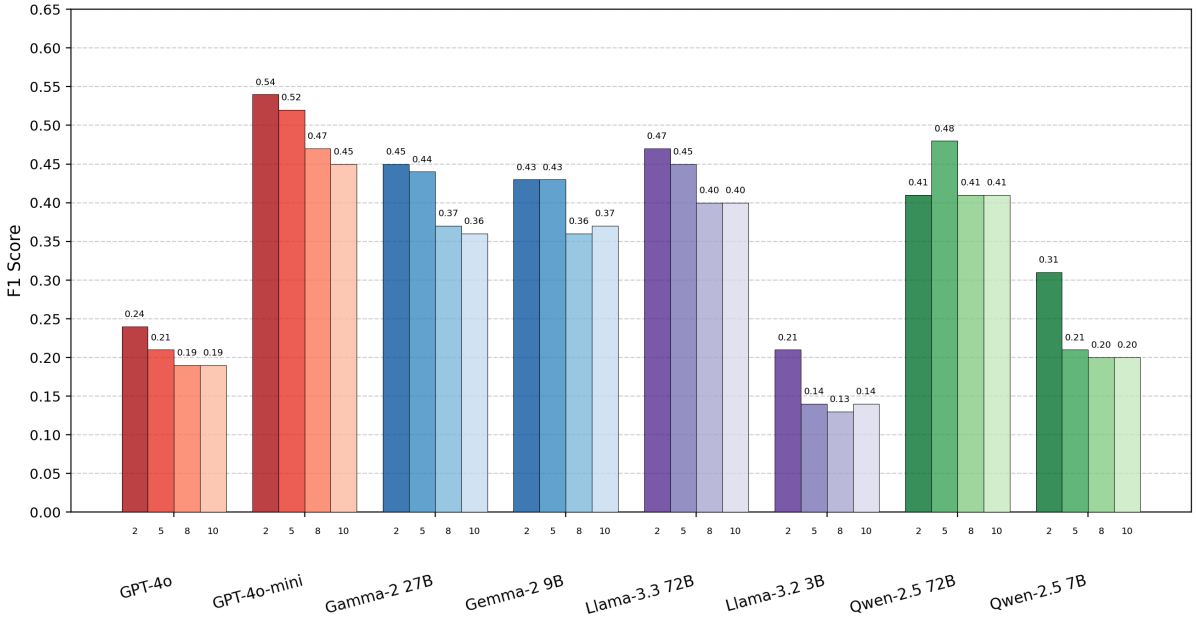
*Equal contribution.

[1] https://github.com/shaharl6000/MoreDocsSameLen

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 19539–19547
November 4-9, 2025



(a) MusiQue



(b) 2WikiMultihopQA

Figure 2: **Increasing the number of retrieved documents can hurt performance.** In retrieval setups with fixed context windows, adding more documents could reduce performance by up to 10 percent. Two models (Llama-3.3 and Gemma-2) showed worse performance, while Qwen-2.5 remained unaffected. The smaller versions of the LLMs (7–9B) show a similar trend as their larger counterparts but the effect is weaker. The hues of the bars represent the amount of retrieved documents.

formance affected by the number of retrieved documents? This disentangles the challenge of long context from the challenge in processing collections of related documents – which often contain redundancies, conflicting information, and implicit inter-document relations (Hirsch et al., 2023; Lior et al., 2024). From a practical perspective, answering this question can help understand a breadth versus depth tradeoff — i.e., whether to strive to retrieve shorter context out of many documents or whether to aim to retrieve longer context out of fewer documents. An ideal experimental setup would have the exact information conveyed *in the same number of tokens* across varying number of

documents, from a long and self-contained single document to a large, multi-document corpus. We find that the custom sets we constructed from MuSiQue (Trivedi et al., 2022) and 2WikiMultiHopQA(Ho et al., 2020), a multi-hop QA datasets, serve as a convenient approximation, allowing us to explore the relationship between long-context and multi-document comprehension in a controlled environment with real-world texts.

Each instance in both datasets consists of a question and a set of documents, where each document is an excerpt from a Wikipedia article retrieved according to the input question. Each instance is constructed such that the question can be answered based on only a subset of the input documents, while the other documents serve as realistic distractors in retrieval settings, as they revolve around the question's topic but do not contain information required to answer the question.

We vary the *number of documents* in the input by gradually removing the distractor documents. When removing a distractor document, we respectively extend each of the remaining documents with distracting content from their corresponding Wikipedia article. Importantly, the process preserves the position of the relevant information within the cotext. This process is illustrated in Fig. 1.

If the context length is the sole challenge, we should expect the performance to remain similar regardless of the number of input documents. Conversely, if processing multiple related documents presents an additional challenge, we would expect an inverse correlation between performance and the number of input documents.

Our evaluation of several state-of-the-art models (Llama-3.3, Qwen2.5, Gemma2, and GPT-4o), presented in Fig. 2, indicates that in most cases, reducing the number of documents while keeping the amount of tokens improves performance by up to 10% in MuSiQue,and up to 20% in 2WMHQA. An exception is Qwen2.5, which may indicate that it better handles multi-document collections.

Our work has several major implications and avenues for future work. First, from a practical perspective, RAG systems should take the number of retrieved documents into consideration, as the introduction of additional documents into the prompt may hurt performance. Second, future work should explore novel approaches for multi-document processing, which according to our findings presents a separate challenge from mere long context. Such work can make use of our paradigm and data for training and evaluation.

2 Multi-Document Evaluation with Controlled Token Count

Our goal is to understand how the number of retrieved documents affects LLM performance when controlling the input length. To this end, we evaluate several models on multi-document multi-hop question answering, which requires models to find relevant information within a given context to answer a specific question. In particular, we make controlled adjustments to the number of documents in the input, while preserving the position of the key information needed to answer the questions, and keeping the context length consistent.

Our datasets are based on MuSiQue (Trivedi et al., 2022) and 2WikiMultiHopQA (Ho et al., 2020), which we nickname as 2WMHQA. Both datasets are multi-hop QA datasets that consist of questions associated with paragraphs (20/10 paragraphs for MuSiQue/2WMHQA) sampled from individual documents, retrieved from Wikipedia according to the question. Of these paragraphs, 2–4 contain the supporting information necessary to answer the question, while the remaining paragraphs serve as realistic distractors in a RAG setup, as they are retrieved from related topics but do not contain relevant information to answer the question. Fig. 1 shows an example query, and a list of retrieved documents, where three are relevant to the question (marked in pink), and the rest are distractors (marked in blue). We further elaborate on the dataset in section A in the appendix.

Leveraging MuSiQue's and 2WMHQA's structure, we constructed several data partitions to investigate the impact of the number of retrieved documents in a controlled manner. The process involved the following steps:

1. **Select the total number of documents:** We reduce the number of documents from the original document count down to only the supporting documents. For MuSiQue from 20 to 15, then 10, 8, and finally down to the 2–4 documents consisting of the relevant information to answer the question. Similarly for 2WMHQA, from 10 to 8, 4, and finally down to the 2 positive documents.

2. **Choose the supporting and non-supporting documents:** We always keep the documents

that support the answer to ensure that the question remains answerable, and randomly select the remaining ones from the non-supporting set. Non-supporting documents remain consistent across different document counts, i.e., each set includes all documents from the smaller sets. Fig. 1 shows such document selection in the two right columns, note that relevant documents (blue) are always kept.

3. **Expand the selected documents:** Since the original documents are Wikipedia paragraphs, we located their source Wikipedia pages and added text preceding and following the paragraphs to match the original token count. This replaces distracting content from different documents with distracting content from the same document. In Fig. 1, we show that each of the remaining documents is expanded to keep the original token count, while ensuring that information from the supporting documents appeared in similar positions across all sets.

3 Evaluation

3.1 Experimental Setup

We evaluated six instruction-tuned LLMs from four model families: Llama-3.3 70B and Llama 3.2 3B (AI@Meta, 2024) [2], Qwen2.5 7B/72B (Qwen et al., 2025), Gemma2 9B/27B (Team et al., 2024), and GPT-4o/GPT-4o-mini (Hurst et al., 2024). Large models were run on Together.ai[3], and smaller ones on an A6000 GPU. We used a decoding temperature of 0.8, as recommended in prior evaluations (Chen et al., 2021). Evaluation relied on overlap F1 between gold and predicted outputs, following MuSiQue (Trivedi et al., 2022). Prompts, formats, and evaluation code were implemented using SEAM (Lior et al., 2024) (see Appendix C for details).

3.2 Results

Our key findings (Fig. 2) reveal that in a retrieval setup, LLMs suffer when presented with more documents, even when the total context length is the same. This may be due to the unique challenges in multi-document processing, which involves processing information that is spread across multiple sources, which can introduce conflicting or overlapping details. Almost all models perform better when presented with fewer documents, with scores improving by 5% to 10% on average in MuSiQue and by 10% to 20% in 2WMHQA. We find that the smaller versions of all LLMs exhibit a similar pattern, albeit to a lesser degree.

An exception is Qwen2.5, which may indicate that it better handles multi-document collections. It performed similarly across the different document quantities in MuSiQue and 2WMHQA.

Interestingly, GPT-4o performed significantly worse than GPT-4o-mini. Recent studies show GPT-4o-mini can outperform GPT-4o on certain tasks (Chen et al., 2024; Nguyen et al., 2025; Alabbasi et al., 2025). This may be because GPT-4o's larger parameter count leads to overfitting, while GPT-4o-mini's smaller size forces it to focus on more generalizable patterns.

Model	No documents	
	MuSiQue	2WikiMultiHopQA
Qwen-2.5 72B	0.01	0.03
Qwen-2.5 7B	0.01	0.02
Llama-3.3 70B	0.05	0.08
Llama-3.2 3B	0.01	0.01
Gemma-2 27B	0.02	0.02
Gemma-2 9B	0.05	0.02
GPT-4o	0.02	0.04
GPT-4o-mini	0.05	0.01

Table 1: F1 scores in a scenario where only the questions are provided (without documents)

3.3 Analysis

To contextualize our results, we created additional versions of our data, discussed below along with the respective findings.

Contamination does not appear to affect our results. To test whether the models relied on memorization, we evaluated them using only the questions, without any retrieved context. All models performed poorly (≈ 0.02 F1), reducing concerns about data contamination. Results in Table 1.

Behavior is similar across instances with different amounts of tokens. We evaluate the performance for instances with different context lengths. Although we keep the number of tokens constant across the different multiplicities of documents, each question and its associated documents have a different token count. To further explore whether there is any difference in performance for instances with different lengths, we check the performance

[2] A small counterpart to Llama-3.3 was not available at the time of evaluation.

[3] https://www.together.ai

as the number of documents increases for different token bins (each bin describes a different range of number of tokens). We observe that for different token bins, the behavior remains the same: as the number of documents increases, the performance degrades. We elaborate in the section B.4 in the appendix.

4 Conclusions

We assess the challenges of multi-document retrieval tasks when varying the number of documents. Our results indicate that input that includes more documents complicates the task in an environment of retrieval settings, highlighting the need for retrieval systems to balance relevance and diversity to minimize conflicts. Future models could benefit from mechanisms to identify and discard conflicting information while leveraging document variety.

5 Limitations

This study does not address prompt variations or the effects of data order within inputs. Future work should explore alternative datasets to ensure more robust evaluations. While our experiments focused on extreme scenarios (highly distracting or random contexts) and document counts between 2–20, future research should investigate more nuanced setups and larger document sets to better reflect real-world conditions. All datasets from this study will be publicly available upon publication for further research in multi-document processing.

References

AI@Meta. 2024. Llama 3 model card.

Nouf Alabbasi, Omar Erak, Omar Alhussein, Ismail Lotfi, Sami Muhaidat, and Mérouane Debbah. 2025. Teleoracle: Fine-tuned retrieval-augmented generation with long-context support for networks. *IEEE Internet of Things Journal*.

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

Yongchao Chen, Harsh Jhamtani, Srinagesh Sharma, Chuchu Fan, and Chi Wang. 2024. Steering large language models between code execution and textual reasoning. *arXiv preprint arXiv:2410.03524*.

Eran Hirsch, Valentina Pyatkin, Ruben Wolhandler, Avi Caciularu, Asi Shefer, and Ido Dagan. 2023. Revisiting sentence union generation as a testbed for text consolidation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7038–7058, Toronto, Canada. Association for Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models.

Gili Lior, Avi Caciularu, Arie Cattan, Shahar Levy, Ori Shapira, and Gabriel Stanovsky. 2024. Seam: A stochastic benchmark for multi-document tasks.

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.

Quoc-Toan Nguyen, Josh Nguyen, Tuan Pham, and William John Teahan. 2025. Leveraging large language models in detecting anti-lgbtqia+ user-generated texts. In *Proceedings of the Queer in AI Workshop*, pages 26–34.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

A Datasets

We use two Multi-Hop QA datasets: MuSiQue (Trivedi et al., 2022) and 2WikiMultiHopQA (Ho et al., 2020), which we nickname as 2WMHQA. Both datasets consist of a set of questions associated with documents mined from Wikipedia. The documents are split between those that contain relevant knowledge to solve the question and distractors that contain similar details but not knowledge that is directly relevant to answering the question.

MuSiQue (Trivedi et al., 2022), a multi-hop QA dataset whose validation set consists of 2,417 answerable questions. Each question is associated with 20 paragraphs sampled from individual documents, retrieved from Wikipedia according to the question. Of these paragraphs, 2–4 contain the supporting information necessary to answer the question, while the remaining paragraphs serve as realistic distractors in a RAG setup, as they are retrieved from related topics but do not contain relevant information to answer the question. The mean token count for questions with their associated documents is 2,400 tokens per instance.

Similarly, 2WMHQA is a multi-hop QA dataset which is composed of questions with associated documents, where only a subset are relevant to answering the question. 2WMHQA's validation set consists of 12,576 answerable questions. Differently from MuSiQue, each question is associated with only 10 paragraphs sampled from individual documents retrieved from Wikipedia. Of these paragraphs, only 2 contain the supporting information necessary to answer the question, while the rest are distractors. In our setup, we choose questions with associated documents that are above 1,500 tokens, which yields a final set of 994 questions. The mean token count for an instance in this dataset is 1,845 tokens.

Model	**Supporting documents only**	
	MuSiQue	**2WikiMultiHopQA**
Qwen-2.5 72B	0.45	0.51
Qwen-2.5 7B	0.23	0.29
Llama-3.3 70B	0.54	0.61
Llama-3.2 3B	0.15	0.20
Gemma-2 27B	0.52	0.57
Gemma-2 9B	0.50	0.53
GPT-4o	0.35	0.20
GPT-4o-mini	0.62	0.65

Table 2: F1 scores for the large and small versions of each model in the scenario only the supporting documents are provided (without expanding the context).

Token Bin	2-4 Docs	8 Docs	10 Docs	15 Docs	20 Docs
0 - 2000	0.45	0.44	0.45	0.41	0.40
2000 - 2500	0.38	0.38	0.35	0.32	0.31
2500 - 3000	0.37	0.35	0.35	0.30	0.28
3000+	0.30	0.29	0.26	0.28	0.28

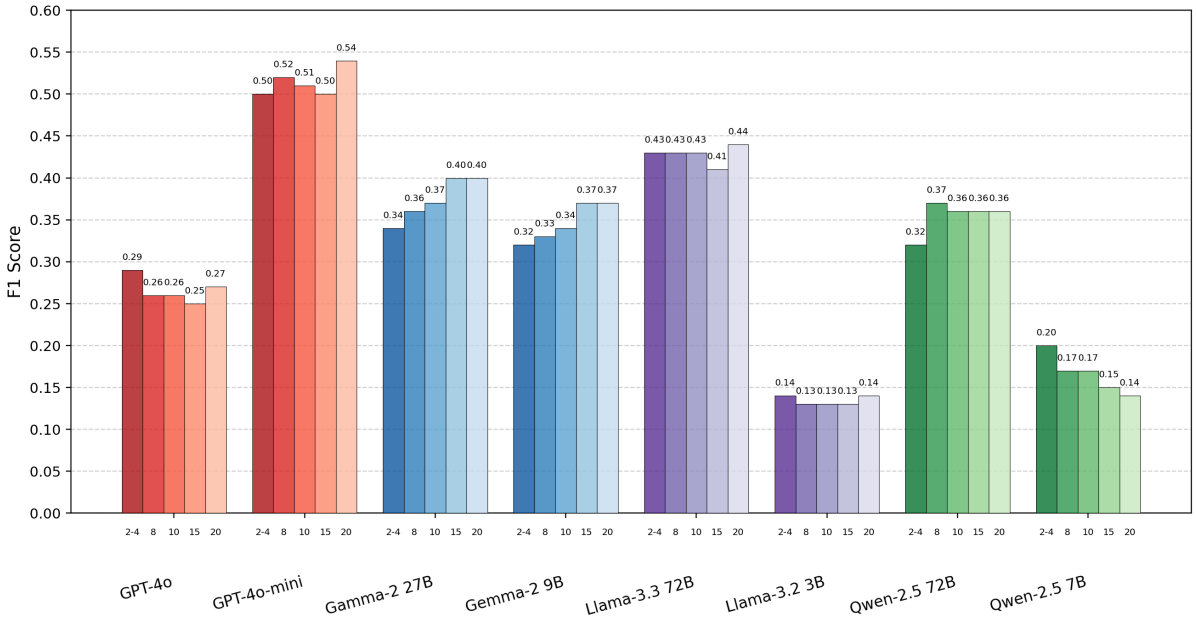
Table 3: F1 scores for Llama-3.1 with 70B parameters performance on MuSiQue. We clustered the different instances in MuSiQue according to the number of tokens. We observe that the model's performance degraded as we increased the number of documents, a pattern that occurred across the different bins.

B Additional Analysis

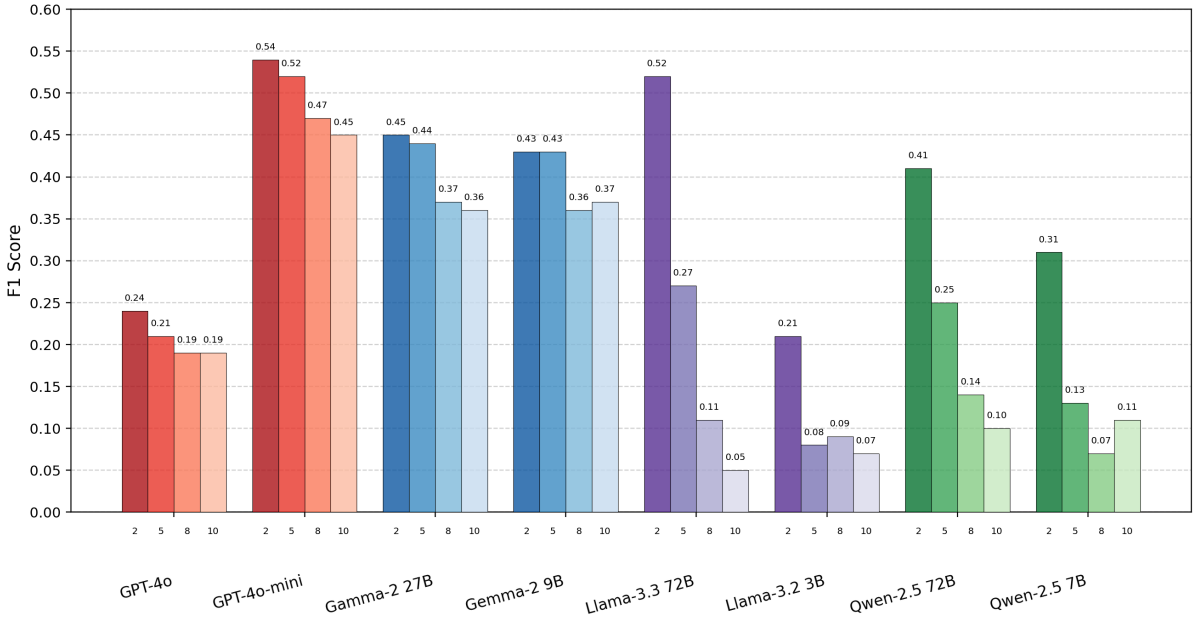
B.1 Random distractors yield inconsistent behavior.

We evaluate all models against versions of the two datasets where we use randomly selected Wikipedia paragraphs instead of retrieved distractors. As shown in Fig. 3, unlike with the original dataset, we observe more nuanced phenomena. For MuSiQue with the large LLM versions, the models' performance improves as more documents with random distractors appear within the input. However, for 2WMHQA, the performance degrades significantly as more random distractors are added. This suggests that the models' behavior varies significantly when using random distractors compared to retrieved ones.

We believe the main reason for the difference in performance between the datasets lies in the positive document length: the retrieval content, although it does not contain the answer, still contains



(a) MusiQue



(b) 2WikiMultihopQA

Figure 3: **The effects of adding non-related documents.** When adding irrelevant documents, LLMs' performance improves across models for MuSiQue while for 2WMHQA it produces significant degradation in performance.

relevant information for answering the question. A positive document in MusiQue is around ten sentences, while a positive document in 2WikiMultihopQA contains only two sentences. Since the positive document is shorter, the model may benefit more from additional knowledge from retrieved documents, even if they do not include the actual answer. Therefore, for MusiQue with longer evidence, the model is less reliant on distracting content, while in 2WikiMultihopQA, where the positive document is only two sentences, the knowledge in the retrieved documents might be crucial.

B.2 Additional context hurts performance.

We test the performance when models are given only the supporting documents, thus providing a much shorter context and eliminating any distracting content. The performance of the LLMs on this set was significantly higher compared to the experimental sets that contained external information.

Full results are shown in Table 2 in the appendix.

B.3 Observed pattern applies to additional variants

We experimented with two additional model variants: Qwen-2 7B/72B(Yang et al., 2024) and Llama-3.1 8B/72B(AI@Meta, 2024). The observed trends remain consistent across different model versions. Qwen-2 demonstrates robustness as document count increases, suggesting it is better suited for multi-document processing, while Llama-3.1 shows a 10% performance decrease, as seen in Figure 4.

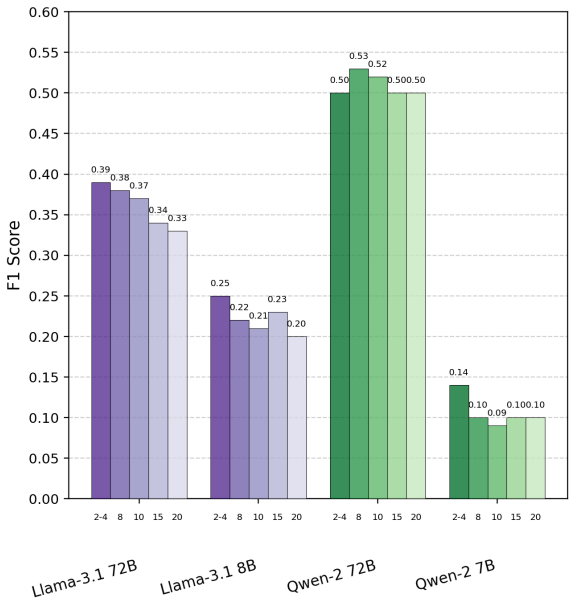
We also tested these variants with random distractors. Both Qwen-2 and Llama-3.1 exhibit similar patterns to their advanced counterparts: performance improves on MuSiQue with random documents, while results on 2WMHQA degrade significantly as document count increases. Results for random distractors can be seen in Figure 5.

B.4 Behavior is similar across instances with different amounts of tokens

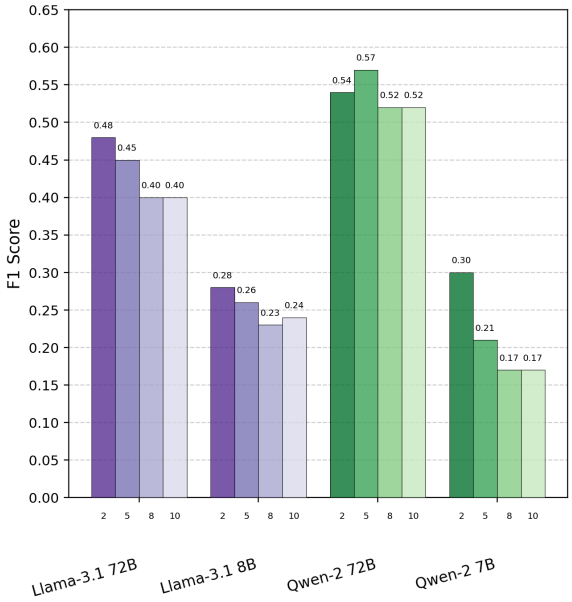
We evaluate the performance of the models for instances with different context lengths. Although we keep the number of tokens constant across the different multiplicities of documents, each question and its associated documents have a different token count. To further explore whether there is any difference in performance for instances with different lengths, we cluster the predictions of Llama-3.1 with 70B parameters on the MuSiQue dataset according to their number of tokens. Then we check the model performance of each cluster across different numbers of documents. We observe that for each token cluster, the performance still degrades independently of the token count. In addition, we observe that the performance is higher when the number of tokens is lower. Results are presented in Table 3 in the appendix.

C prompt

We use prompt 1 for all model and document quantities taken from SEAM (Lior et al., 2024).

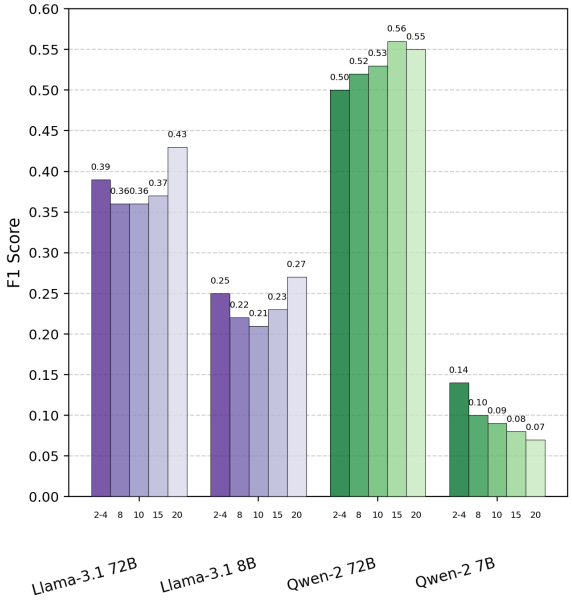


(a) MusiQue

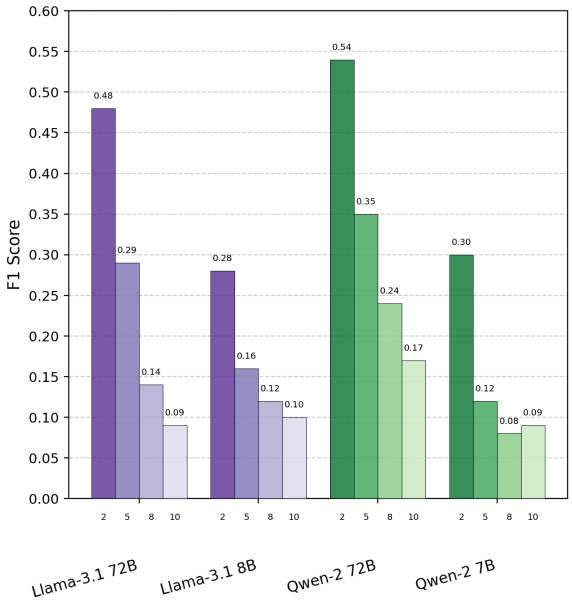


(b) 2WikiMultihopQA

Figure 4: **Performance of previous model variants with increasing retrieved documents.** We tested earlier model versions (Llama-3.1 and Qwen-2) in retrieval settings with fixed context windows while adding more documents. Our findings were consistent with the latest model versions. Llama-3.1 showed performance reductions of up to 10%, similar to Llama-3.3, while Qwen-2 remained unaffected, consistent with Qwen-2.5's behavior.



(a) MusiQue



(b) 2WikiMultihopQA

Figure 5: **The effects of adding non-related documents for previous variants.** Similarly to the latest variants, when adding irrelevant documents, the LLMs' performance improves across models for MuSiQue while for 2WikiMultiHopQA it produces significant degradation in performance.

```
In this task, you are presented with a question
    and 20 documents that contain information
    related to the question. Your goal is to
    deduce your answer solely from the provided
     documents. You must not use any external
    data sources or prior knowledge.

- Carefully read and analyze each document.
- Identify relevant information to accurately
    answer the question.
- Formulate a short, concise, and precise answer.

- Exclude irrelevant details from your answer.

Output format:
Return your answer in the following JSON
    dictionary structure:

- If the provided documents contain the answer:
{
  "is_answerable": true,
  "answer_content": "Your concise answer derived
     directly from the documents."
}

- If the provided documents do NOT contain
    sufficient information to answer the
    question:
{
  "is_answerable": false
}

Important:
- Ensure that your answer strictly adheres to
    the information in the provided documents.
- Do not include speculation, external facts, or
     personal interpretations.
```

Listing 1: Inference prompt