RAGGED: Towards Informed Design of Retrieval Augmented Generation Systems

Jennifer Hsia* **Afreen Shaikh*** **Zhiruo Wang** **Graham Neubig**
Carnegie Mellon University
{jhsia2,afreens,zhiruow,gneubig}@cs.cmu.edu

Abstract

Retrieval-augmented generation (RAG) systems have shown promise in improving task performance by leveraging external context, but realizing their full potential depends on careful configuration. In this paper, we investigate how the choice of retriever and reader models, context length, and context quality impact RAG performance across different task types. Our findings reveal that while some readers consistently benefit from additional context, others degrade when exposed to irrelevant information, highlighting the need for tuning based on reader sensitivity to noise. Moreover, retriever improvements do not always translate into proportional gains in reader results, particularly in open-domain questions. However, in specialized tasks, even small improvements in retrieval can significantly boost reader results. These insights underscore the importance of optimizing RAG systems by aligning configurations with task complexity and domain-specific needs. [2]

1 Introduction

RAG [Chen et al., 2017, Lewis et al., 2020] is widely applied to enhance the performance of top-performing LMs on knowledge-intensive generation tasks like DBQA [Karpukhin et al., 2020]. Given a question, the *retriever* model retrieves multiple relevant passages from a corpus, which are then included as context for the reader model to generate a grounded response.

Although using RAG supposedly helps LMs generate "more specific and factually accurate responses" [Lewis et al., 2020], we show that, in practice, achieving the greatest benefits from RAG requires careful configuration of all components in the RAG pipeline. Existing literature provides mixed, even contradictory, suggestions for configuring RAG. While some early works suggest that providing more retrieved passages results in strictly better outputs [Izacard and Grave, 2021], others find there is a limit to that phenomenon as model performance saturates after some number of contexts [Liu et al., 2023]. Others find that reader model performance declines [Cuconasu et al., 2024, Jiang et al., 2024] as the number of contexts gets too large. The complexity of choosing the number of passages is only one aspect of RAG configuration among many that we cover in our analysis framework.

To provide more concrete suggestions of the *best practices* under various cases, we introduce an analysis framework, RAGGED,[3] study RAG configurations on a suite of representative document-based question-answering (DBQA) tasks, including open-domain datasets that are single-hop and multi-hop questions [Kwiatkowski et al., 2019, Yang et al., 2018], and special-domain questions from the biomedical domain. We cover a broad range of models to ensure a comprehensive analysis: for retrievers, we incorporate both sparse and dense retrievers; for readers, we cover proprietary API models such as GPT [Brown et al., 2020] and CLAUDE [Enis and Hopkins, 2024], as well as

* Equal contribution.
[2] Code/data for the RAGGED framework are available at https://github.com/neulab/ragged
[3] For "**R**etrieval **A**ugmented **G**eneration **G**eneralized **E**valuation **D**evice".

Preprint. Under review.

open-checkpoint models including FLAN [Chung et al., 2022, Tay et al., 2023], LLAMA [Touvron et al., 2023b] families.

In this paper, we address the following key research questions:

R1: When does RAG improve performance over closed-book generation?(§3) We explore whether RAG consistently enhances reader performance across different reader models and datasets. This analysis seeks to identify the specific scenarios — such as particular readers or question types — where RAG provides a clear advantage over closed-book generation, or whether its benefits are more situational.

R2: How do reader models respond to an increasing number of context documents?(§4) We investigate how reader performance is affected by the amount of context provided. Specifically, we examine whether adding more context passages improves model accuracy, leads to diminishing returns, or even degradation in performance due to too much noisy information (Figure 1).

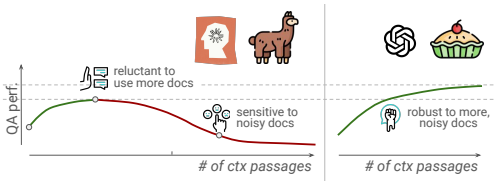


Figure 1: Example insight from using RAGGED: LLAMA and CLAUDE models are more sensitive to noise in context, while FLAN and GPT models are more robust to noise in context and can effectively use a larger number of context passages.

R3: How robust are reader models to irrelevant information when relevant information is present or absent?(§5) We assess how reader models perform on data slices where relevant information is present and on slices where it is absent. This analysis is crucial for understanding model robustness to irrelevant information.

R4: How does retriever choice impact reader performance across question types and domains?(§6) To understand the impact of context quality from another perspective, we evaluate the effect of retriever model choice across different question types. This investigation aims to identify the retriever-reader combinations that yield the best results depending on the task and domain.

In summary, our study provides actionable insights into when and how RAG can be effectively applied, offering guidance for configuring RAG systems to maximize their advantages.

2 The RAGGED Framework

In our analysis, we vary three key aspects:

1. **RAG system components:** For retrievers, we use two approaches:(1) BM25 [Robertson et al., 2009], a sparse retriever based on lexical information, and (2) ColBERT [Santhanam et al., 2021], a dense retriever based on neural embeddings. For readers, we examine both closed-source models from the GPT and CLAUDE families, and open-source models from the FLAN, LLAMA2, and LLAMA3 families.
2. **Number of retrieved passages (k):** We vary the number of retrieved passages from 1 to 50, with most insightful variations occurring before k = 30.
3. **Data slices based on retrieved passage quality:** Passage quality refers to the presence of "gold" passages in the top-k retrieved set.

For datasets, we adopt three DBQA datasets from two domains (Wikipedia and biomedical) and with varying complexity (single-hop, multi-hop). These include **Natural Questions (NQ)** [Kwiatkowski et al., 2019] for open-domain, single-hop questions; **HotpotQA** [Yang et al., 2018], a Wikipedia-based dataset requiring reasoning across multiple passages; and **BioASQ Task 11B** [Krithara et al., 2023], a single-hop PubMed-based biomedical dataset. Details about the corpus of passages used for retrieval are in Table 4. More details about the models and datasets can be found in the subsection A.2.

For the metrics, we follow [Petroni et al., 2021] to evaluate retrieval performance using **recall@k**, which measures the fraction of ground-truth passages among the top-k retrieved. Reader performance is assessed using **unigram F_1**, which measures the unigram overlap between the reader's output and the gold answer. For each query, the highest F_1 score of the generated answer against the list of gold answers is reported.

3 When Does RAG Surpass the No-Context Baseline?

While Lewis et al. [2020] achieve state-of-the-art results across several QA tasks by augmenting T5 model with a fixed k number of documents, we find that the answer to **RQ1**: "When does RAG

outperform no-context baseline" is more nuanced. Below, we share our findings: Some retriever-reader combinations consistently benefit from RAG, while others only sometimes do for certain k's, and a few never do regardless of k.

Although GPT-3.5 generally performs better with RAG, the gains are small at around 1.1 F1 points on average. In contrast, FLAN models consistently use contexts effectively and outperform their no-context baseline across retrievers, k, and datasets. In fact, FLANT5 can use retrieved contexts so well that even though it ranks among the bottom-3 readers in terms of no-context performance, it ranks among the top-3 models in terms of optimal-k performance. Although LLAMA2 often benefit from RAG, they usually do so only when k is small enough. Then, finally, LLAMA3 and CLAUDE HAIKU struggle to benefit from RAG, often performing with context rather than without regardless of retriever and k.

Key Takeaway: The effectiveness of RAG varies with reader choice and sensitivity to context quantity and quality. The results suggest that while RAG can help specific models, the degree of improvement varies, and in some cases, models perform worse with retrieved contexts.

4 Are More Contexts Always Better?

To address **RQ2**, we evaluate whether adding more retrieved passages consistently improves reader model performance. While Liu et al. [2023] report that reader performance saturates as k increases, Cuconasu et al. [2024] and Jiang et al. [2024] observe performance degradation with increasing k. Although these findings appear contradictory, we argue that they are actually complementary, as each study focuses on a limited range of retrievers, readers, and datasets. Our experiments, which span a wider variety of retrievers, readers, and datasets, demonstrate that both saturation and degradation behaviors can occur *with the determining factor being the choice of reader model.*

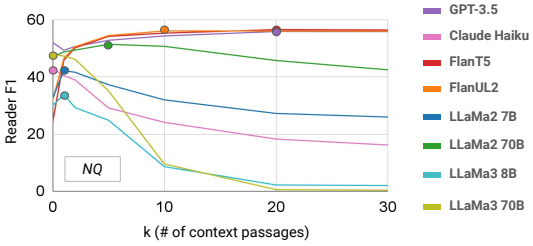


Figure 2: Reader performance as we vary k, the number of retrieved contexts provided by ColBERT, from 0 (no context) to 20. Colored circles mark the reader performance at optimal k^*.

Figure 2 reveals two typical trends in reader performance as k, the number of context passages, increases. Some readers effectively identify and use signals, thus displaying an improve-then-plateau trend as k increases. Such models include the FLAN models and GPT-3.5, and they often peak at $k \geq 10$ without noticeable decline afterward.

In contrast, other models are more easily distracted by "noise", or irrelevant context, as the number of contexts increases, thereby displaying a peak-then-decline behavior. Such models include LLAMA and CLAUDE HAIKU, and they generally peak early at $k < 5$ before degrading in performance.

Key Takeaway: Optimizing RAG performance depends on selecting the suitable reader model and adjusting k— while some models are more sensitive to signal, benefitting from larger k values, others are more sensitive to noise, requiring smaller k.

5 Reader Response to Signal and Noise

To address **RQ3**, we study how the readers respond to noise under contexts with at least one "signal" (gold) passage and without any "signal". In the first scenario (§5.1), where sufficient information is available to answer the question, we compare reader performance across three settings: (a) with the top-k passages, (b) with only the gold passages among the top-k (top-gold), and (c) with no context. We compare how big the gap between (a) and (b) is to see how close the top-k performance is to the performance when it focuses only on the signal. We also compare (a) and (c) to see how detrimental noise is — is the noise so distracting that (a) is even worse than the no-context baseline (c)?

In the second scenario (§5.2), where there is insufficient context to answer the question, we assess the model's ability to filter out noise and rely on its pre-trained knowledge.

5.1 With Gold Passages

Reader models with only gold passages expectedly serve as an upper bound for their top-k performance (Figure 3). However, it is notable that *the no-context performance does not always represent*

a lower bound for RAG. Whether it is a lower bound depends on the reader's ability to filter out irrelevant information while leveraging helpful context information.

For NQ instances with signal in the top-k passages, GPT-3.5 and FLAN models effectively identify and use relevant information, consistently outperforming no-context baselines (Figure 3). In contrast, the rest of the models struggle more with noise, with their top-k performance falling below their no-context results at $k \leq 5$. This shows that *suboptimal configurations can lead to worse performance with RAG, even when sufficient information is available*.

In HotpotQA, the LLAMA2 models maintain performance above the no-context baseline longer than in NQ, with LLAMA2 7B dipping below at $k = 25$ instead of at $k = 15$ and LLAMA2 70B dropping below the no-context baseline at $k \geq 30$ instead of at $k = 25$ (Figure 6). Similarly, CLAUDE drops below the baseline at $k > 5$ instead of $k \leq 5$. This could suggest that *tasks requiring multiple signal passages provide more "anchor points" for the model, helping it withstand more noise*(Figure E).

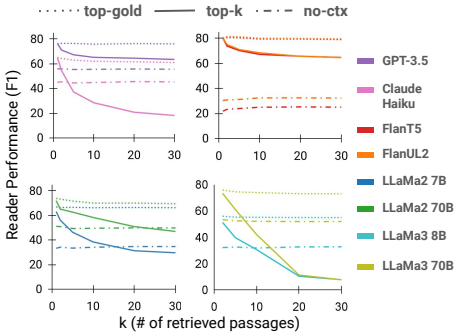


Figure 3: NQ results when there is sufficient information (at least one gold passage) in the top-k passages to answer the question. Top-gold means the context only includes the gold passages in the top-k passages.

For BioASQ, *all readers' gaps between their top-pos and top-k performances are smaller than their gaps on open-domain datasets* (Figure 7), indicating better signal extraction likely due to the specialized domain jargon making relevant documents more distinct. We attribute the smaller gap primarily to the reader instead of the retriever since the retrieval quality for BioASQ is strictly worse than NQ (Table 5). Also of note is that CLAUDE HAIKU and LLAMA3 70B still fall below their no-context baselines even with gold passages, showing that they struggle particularly with specialized domains. In these cases, the models often generate nonsensical outputs as k increases.

5.2 Without Gold Passages

We conduct a similar analysis with examples retrieved with only non-gold passages. For NQ (Figure 4) and HotpotQA (Figure 8), most models perform worse with RAG than without. This is expected since these slices the models are prompted to rely on do not contain any signal (i.e., gold passages).

In contrast, FLAN *models consistently outperform their no-context baselines even with non-gold contexts*. One potential explanation is that the non-gold passages may still provide partially relevant information despite insufficient information.

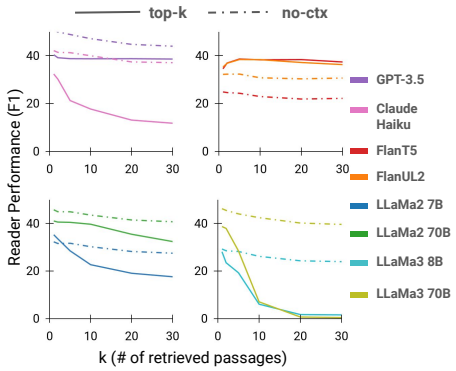


Figure 4: NQ results with no gold passages.

For BioASQ, a key difference is that GPT's top-k performance exceeds its no-context baseline for $k \geq 5$, and LLAMA2 7B's performance does so for all k, unlike their consistently lower performance in other datasets. This suggests that for specialized-domain questions, these models may have stronger guardrails against irrelevant information and can rely on their pre-trained knowledge when needed. Full results for other models are in Figure 9

Key Takeaway: Practitioners can ensure stable performance in noisy environments by using robust models or applying noise-filtering techniques when models are sensitive to irrelevant information.

6 Impact of Retriever Choice

To address RQ4, we compare BM25 and Colbert retrieval performance (Table 5) and analyze their impact on reader model performance (Figure 2, Figure 5b).

Neural retrievers like ColBERT generally outperform lexical retrievers such as BM25, but the extent of this advantage on reader performance varies by domain and question complexity.

We evaluate the **average difference**, which is the mean difference between F_1 scores when the reader is paired with top-k documents from ColBERT v. BM25, averaged across $k = 1$ to 50 (Table 2). Although ColBERT offers significant retriever recall gains over BM25 for open-domain datasets (21.3 for NQ and 14.6 for HotpotQA Table 5), the corresponding average reader performance gains are modest (5.2 and 1.9 F1 points, respectively). In contrast, for BioASQ, despite a smaller recall gain (0.7), the reader performance gain is relatively larger (1.5 F1 points). This suggests that *in specialized domains, even a small improvement in retriever performance can have an outsized impact on reader results.*

Model	NQ	HotpotQA	BioASQ
GPT−3.5	8.6	2.0	1.1
Claude Haiku	3.9	4.0	2.4
FlanT5	12.6	10.5	4.2
FlanUL2	12.9	2.0	1.9
LLaMa2 7B	3.6	0.9	−0.3
LLaMa2 70B	2.6	0.7	−0.2
LLaMa3 8B	−0.7	−2.2	1.4
LLaMa3 70B	−1.9	−2.7	1.5
Average	**5.2**	**1.9**	**1.5**

Table 1: Difference in reader performance (F1 score) when using ColBERT vs. BM25, averaged over $k = 1$ to 50.

Given how 1) ColBERT only results in small optimal reader gains for HotpotQA and BioASQ and 2) BM25 is less computationally expensive to use, it may be tempting to claim BM25 is the obvious pick for RAG, computationally speaking. However, another important factor to consider is the difference in optimal k — *the optimal k with BM25 performance is 2 to 3 times that of the optimal k for ColBERT* (Table 6). This means BM25's higher k shifts the computational burden from the retriever to the reader, where the cost of inference is scaled with k.

Key Takeaway: Retriever improvements do not always lead to better reader performance. Practitioners should carefully evaluate both components together, especially for complex open-domain questions, where significant gains in retriever recall may not yield proportional benefits for the reader. In contrast, even minor retriever improvements can significantly improve reader performance for domain-specific tasks.

Retriever	NQ	HotpotQA	BioASQ
BM25	10.95	36.94	23.03
Colbert	28.0	50.68	23.83
Difference	17.05	13.74	0.7

Table 2: Retriever recall (recall@k) across different datasets averaged over $k = 1$ to 50.

7 Conclusion

We propose RAGGED, a framework designed to assist researchers and practitioners in making informed decisions about designing RAG systems, focusing on three key aspects: the number of contexts, the reader model, and the retriever model. Our findings show that RAG systems' effectiveness depends on careful configuration—some readers benefit from additional context, while others degrade with irrelevant information. Retriever improvements don't always translate to better reader performance, especially in open-domain tasks, though small gains can significantly impact specialized tasks. These results emphasize the need for task-specific RAG tuning and further research on enhancing retriever-reader interactions and noise robustness. We hope our framework will help the community better understand how to customize RAG systems to unlock its full potential.

Acknowledgements

Special thanks to Alex Cabrera, Alex Bäuerle, Jun Araki, Md Rizwan Parvez for providing Zeno support for analysis visualization. Our appreciation extends to Hao Zhu, Jacob Springer, and Vijay Viswanathan for providing feedback for our paper. This paper was supported in part by a gift from Bosch research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott

Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729, 2024.

Maxim Enis and Mark Hopkins. From llm to nmt: Advancing low-resource machine translation with claude, 2024.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2021.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*, 2024.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10:170, 2023. URL https://doi.org/10.1038/s41597-023-02068-4.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019. URL https://aclanthology.org/Q19-1026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

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

National Library of Medicine. Pubmed baseline 2023 repository, 2023. URL https://lhncbc.nlm.nih.gov/ii/information/MBR.html.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2023.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*, 2021.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2023.

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. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.

A Related Work

A.1 Prompt

We use the following prompt, and provide more details at §B.

A.2 Setup details

BM25 BM25 is a probabilistic retrieval model [Robertson et al., 2009] that estimates passage relevance via term weighting and passage length normalization. BM25 relies on term-matching, and thus is supposed to be proficient at identifying lexical similarity especially in special domains.

ColBERT One of the best-performing neural-based retrievers is ColBERT [Santhanam et al., 2021], i.e., contextualized late interaction over BERT. ColBERT uses contextualized embeddings instead of term-matching as in BM25, thus is supposed to be better at identifying semantic similarities between queries and passages.

FLAN The FLAN models are encoder-decoder models. We use the FLANT5-XXL [Chung et al., 2022] with 11B parameters and FLAN-UL2 [Tay et al., 2023] with 20B parameters, both with a context length of $2k$ tokens. FLANT5-XXL is an instruction-tuned variant of the T5 model [Raffel et al., 2023].

LLAMA We use 7B and 70B LLAMA2 models [Touvron et al., 2023a,b] and the 8B and 70B LLAMA3 models. The LLAMA2 models have a context length of 4k tokens while LLAMA3 models have double the context length at 8k tokens.

GPT We use GPT-3.5-turbo model [Brown et al., 2020]. This model has a context length of 16k tokens, and is a closed source model, so further details about model size are unknown.

CLAUDE We use CLAUDE HAIKU, which is Anthropic's fastest and most compact model [Enis and Hopkins, 2024]. The context window of 200k tokens is the largest of all the models we compare in this paper, but the model size is unknown since the model is closed-source.

B Implementation Details

Reader model We truncate the *Context* to make sure the the rest of the prompt still fits within a reader's context limit. Specifically, when using FLANT5 and FLANUL2 readers, we use T5Tokenizer to truncate sequences to up to $2k$ tokens; when using LLAMA models, we apply the LlamaTokenizer and truncate sequences by $4k$ tokens for LLAMA2 and $8k$ for LLAMA3. For closed-source models, we spent around $300. Subsequently, we incorporate a concise question-and-answer format that segments the query using "Question:" and cues the model's response with "Answer:", ensuring precise and targeted answers.

For our reader decoding strategy, we used greedy decoding with a beam size of 1 and temperature of 1, selecting the most probable next word at each step without sampling. The output generation was configured to produce responses with 10 tokens. The experiments were conducted on NVIDIA A6000 GPUs, supported by an environment with 60GB RAM. The average response time was $\sim$1.1s per query when processing with a batch size of 50.

C Dataset Details

All corpus and datasets use English.

For NQ and HotpotQA datasets in the open domain, we use the Wikipedia paragraphs corpus provided by the KILT benchmark [Petroni et al., 2021].

For BioASQ, we use the PubMed Annual Baseline Repository for 2023 [of Medicine, 2023], where each passage is either a title or an abstract of PubMed papers. Dataset sizes are in Table 4.

Corpus	# of par	# of doc	Avg # of doc
Wikipedia	111M	5M	18.9
Medline	58M	34M	1.7

Table 3: Retrieval corpus information

The Medline Corpus is from of Medicine [2023] provided by the National Library of Medicine.

For NQ and HotpotQA, we use KILT's dev set versions of the datasets, allowed under the MIT License [Petroni et al., 2021]. For BioASQ [Krithara et al., 2023], we use Task 11B, distributed under CC BY 2.5 license.

Dataset	# of Queries
NQ	2837
HotpotQA	5600
BioASQ	3837

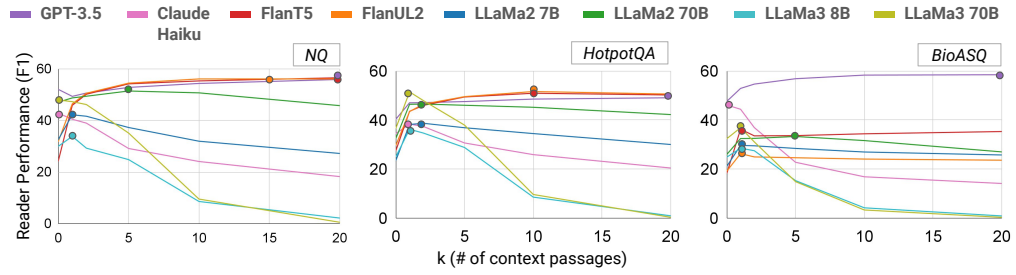
Table 4: Dataset information

D Retriever Performance

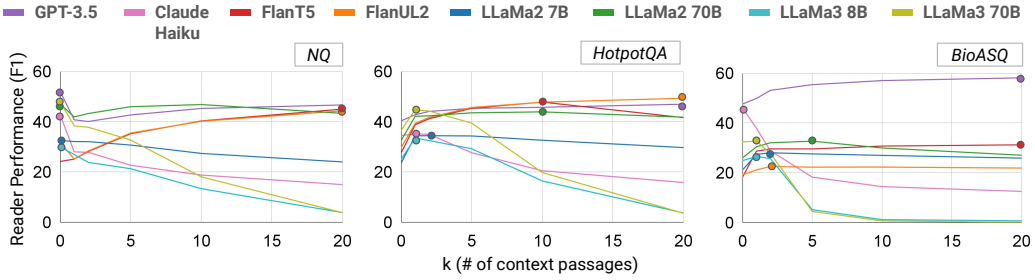
Retriever	Recall@k					
	1	2	5	10	20	50
NQ						
BM25	2.7 10.3	4.4 16.3	8.0 27.8	11.5 36.8	16.3 47.7	22.8 53.2
ColBERT	12.3 27.2	18.0 38.8	25.7 54.4	32.1 65.0	38.1 72.9	41.8 77.2
HotpotQA						
BM25	19.1 23.3	25.9 31.2	34.6 42.7	41.1 52.1	46.8 59.1	54.2 62.8
ColBERT	31.1 34.2	40.1 44.7	49.9 56.3	56.2 63.6	61.9 69.9	64.9 73.1
BioASQ						
BM25	8.8 12.4	12.9 16.4	19.6 23.9	25.8 30.6	33.3 38.7	37.8 43.6
ColBERT	8.8 14.2	13.5 18.2	20.7 25.6	27.1 32.2	34.3 39.8	38.6 44.2

Table 5: For the Wikipedia-based dataset, the top row indicates recall@k at the retrieval unit of Wikipedia paragraph and the bottom row for the unit of Wikipedia page. For BioASQ, the top row indicates recall@k at the unit of title or abstract of a PubMed article and the bottom row at the unit of the article itself.

E Additional Reader Results



(a) Performance when top-k passages are from ColBERT.



(b) Performance when top-k passages are from BM25.

Figure 5: Top-k performance on NQ, HotpotQA, and BioASQ. Colored circles mark the reader performance at optimal k^*. We find similar reader trends to increasing context regardless of the retriever choice.

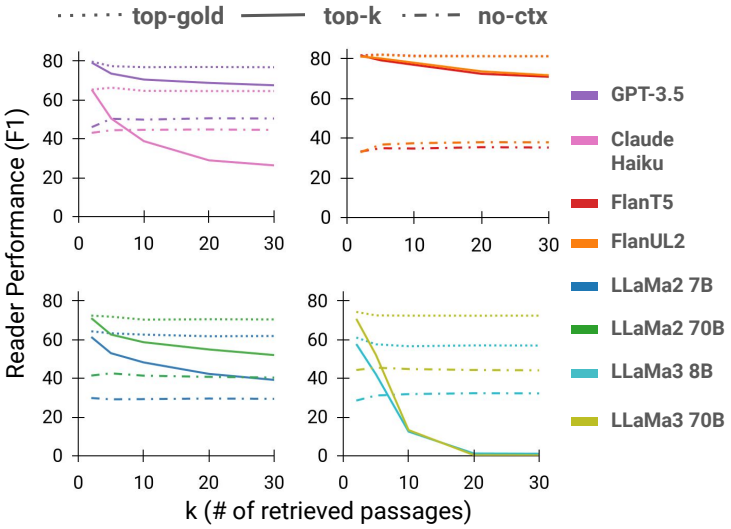


Figure 6: HotpotQA results when there is sufficient information (all gold passages) included in the top-k passages to answer the question. For multi-hop questions, we select examples retrieved with all gold passages within the top-k passages since all passages are necessary to answer the question.

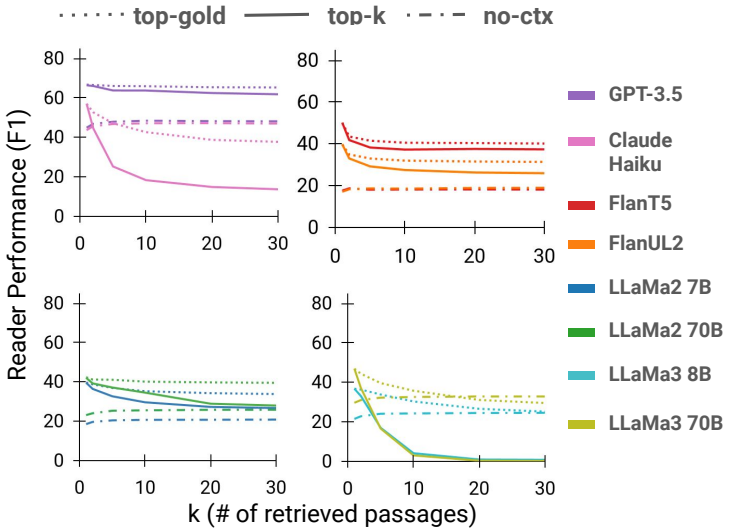


Figure 7: BioASQ results when there is sufficient information (at least one gold passage) included in the top-k passages to answer the question.

Model	NQ		HotpotQA		BioASQ		Average (per reader)	
	BM25	ColBERT	BM25	ColBERT	BM25	ColBERT	BM25	ColBERT
GPT-3.5	50	20	50	20	20	20	40	20
Claude Haiku	1	1	1	1	1	1	1	1
FlanT5	50	20	10	10	50	1	36.67	10.33
FlanUL2	50	10	20	10	2	1	24	7
LLaMa2 7B	1	1	2	2	2	1	1.67	1.33
LLaMa2 70B	10	5	10	2	5	5	8.33	4
LLaMa3 8B	1	1	1	1	1	1	1	1
LLaMa3 70B	1	1	1	1	1	1	1	1
Average (per dataset)	20.5	7.38	11.88	5.88	10.25	3.88	14.21	5.71

Table 6: Optimal k^* for BM25 and ColBERT (NQ, HotpotQA, and BioASQ).

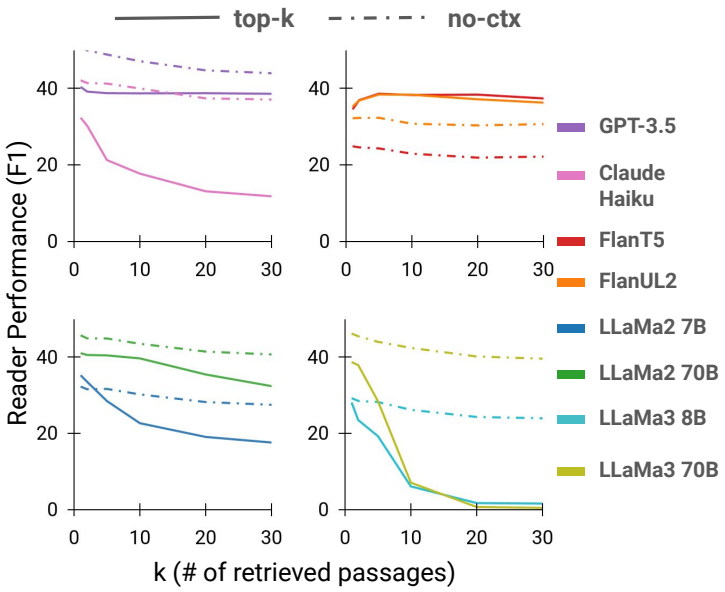


Figure 8: HotpotQA results when there are no gold passages included in the top-k passages to answer the question.

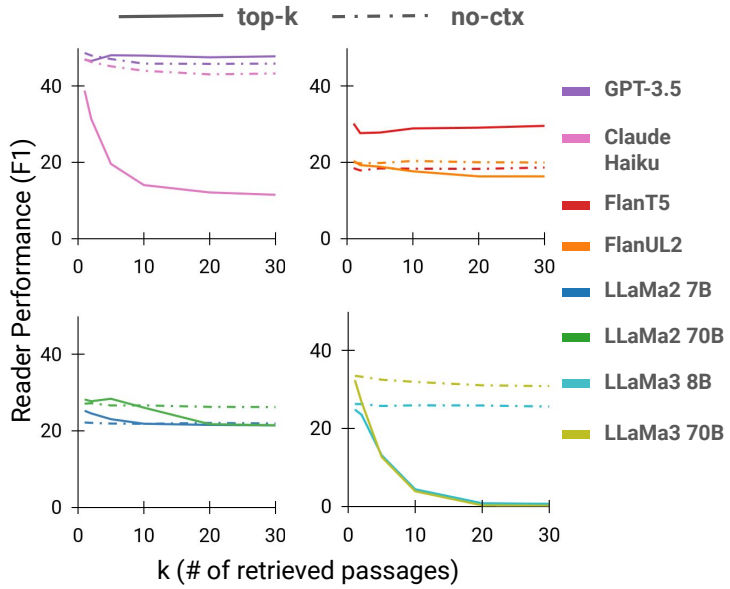


Figure 9: BioASQ results when there are no gold passages included in the top-k passages to answer the question.