npj | digital medicine

Article

Published in partnership with Seoul National University Bundang Hospital

https://doi.org/10.1038/s41746-025-01651-w

Leveraging long context in retrieval augmented language models for medical question answering

Gongbo Zhang[1], Zihan Xu[2], Qiao Jin[3], Fangyi Chen[1], Yilu Fang[1], Yi Liu[4], Justin F. Rousseau[5,6], Ziyang Xu[7], Zhiyong Lu[3], Chunhua Weng[1] ✉ & Yifan Peng[2] ✉

While holding great promise for improving and facilitating healthcare through applications of medical literature summarization, large language models (LLMs) struggle to produce up-to-date responses on evolving topics due to outdated knowledge or hallucination. Retrieval-augmented generation (RAG) is a pivotal innovation that improves the accuracy and relevance of LLM responses by integrating LLMs with a search engine and external sources of knowledge. However, the quality of RAG responses can be largely impacted by the rank and density of key information in the retrieval results, such as the "lost-in-the-middle" problem. In this work, we aim to improve the robustness and reliability of the RAG workflow in the medical domain. Specifically, we propose a map-reduce strategy, BriefContext, to combat the "lost-in-the-middle" issue without modifying the model weights. We demonstrated the advantage of the workflow with various LLM backbones and on multiple QA datasets. This method promises to improve the safety and reliability of LLMs deployed in healthcare domains by reducing the risk of misinformation, ensuring critical clinical content is retained in generated responses, and enabling more trustworthy use of LLMs in critical tasks such as medical question answering, clinical decision support, and patient-facing applications.

Large language models (LLMs) are finding their way into an expanding range of healthcare domains, holding tremendous potential for improving patient care, enhancing communication and education, and facilitating clinical workflow effectiveness[1–7]. LLMs are useful for answering common queries related to diseases or personal risk, interpreting laboratory results, and getting advice on medical condition management[8–12]. Despite the potential of LLMs, the deployment of LLMs in healthcare faces significant safety threats. LLMs struggle to generate accurate and up-to-date responses on current topics, due to outdated knowledge, lack of domain-specific expertise, or hallucination[13–17].

Retrieval-Augmented Generation (RAG) is a pivotal innovation to enhance the quality and relevance of responses in LLMs[18–21]. Typically, a RAG system consists of a retrieval module and a generative module. When a user query is provided as input, the system first uses the retrieval module to fetch relevant documents or data snippets by searching through external data sources. Next, the generative module takes the retrieved information as input and produces a response to the user query. With the help of the retrieval module, the generative module can provide more accurate and factual answers without the need for continual training or fine-tuning. As such, RAG poses a promising direction for applications requiring high factual accuracy and specificity[14,22].

However, prompting LLMs with contextual information has trade-offs. On the one hand, providing contextual information enhances the model's ability to perform the downstream tasks by augmenting LLMs with external domain-specific knowledge that is under-represented in their pretraining data. On the other hand, the input of LLMs is bounded by the limit of their context windows. Even though recently released models can process an increasing number of tokens, the increased amount of content to reason over can still hinder model performance[23]. The quality of RAG completion also depends on the retrieval results, such as the density or positions of query-relevant information[14,22,24–26]. As retrieval systems are still imperfect, it is inevitable to retrieve information irrelevant to the user query[14].

[1]Department of Biomedical Informatics, Columbia University, New York, NY, USA. [2]Department of Population Health Sciences, Weill Cornell Medicine, New York, NY, USA. [3]Division of Intramural Research, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA. [4]Division of Endocrinology, Department of Medicine, Diabetes and Metabolism, Weill Cornell Medical College, New York, NY, USA. [5]Department of Neurology, University of Texas Southwestern Medical Center, Dallas, TX, USA. [6]Peter O'Donnell Jr. Brain Institute, University of Texas Southwestern Medical Center, Dallas, TX, USA. [7]Department of Dermatology, NYU Grossman School of Medicine, New York, NY, USA. ✉e-mail: cw2384@cumc.columbia.edu; yip4002@med.cornell.edu

A recent study reports an issue of "lost-in-the-middle", i.e., the position of key information in the LLM context impacts the quality of the model completions[24]. This issue occurs when a lengthy context of information is retrieved, and the highly relevant information is not ranked at the top or bottom of the retrieval results. We refer to these positions as spotlight positions, and the document containing key information as the key document. If not ranked at the top, the key information may be neglected by the generative module, resulting in incomplete or inaccurate responses to the user queries[24]. How to effectively utilize contextual information in RAG applications remains to be an open research question. Current studies attribute this issue to positional attention bias, i.e., more attention weights are allocated more to information at spotlight positions than others[27–29]. To address the issue, existing methods mainly focus on adjusting the model weights, either by fine-tuning LLMs[27] or directly adjusting the attention weights[28,29]. However, adjusting model weights can lead to catastrophic forgetting[30,31], i.e., the overall performance of LLMs degrades upon adopting new information on a specific task.

In this research, we aim to address the "lost-in-the-middle" issue without modifying model weights. Our strategy involves increasing the density of key information within the context, rather than modifying model weights. The lower bound for RAG is closed-book settings, where LLMs have access only to the question with no extra information. The upper bound is Oracle settings, where only relevant key information is provided in the context. Compared to closed-book settings, LLMs perform significantly better in Oracle settings. These two scenarios represent opposite ends of the spectrum concerning key information density. In closed-book settings, the density is essentially zero because no external information is provided. In contrast, Oracle settings boast nearly 100% density, as only relevant information is supplied. RAG sits in the middle, where relevant information is often mixed with irrelevant content. We hypothesize that the density of key information affects downstream model performance.

Therefore, we propose a novel framework, BriefContext, to transform the long-context reasoning task into multiple short-context reasoning tasks. The core of the framework leverages the map-reduce concept[32–37], originally designed for processing massive data in parallel. In our workflow, we divide the long context into multiple partitions and dispatch them to multiple LLM sessions. The additional LLM service requests incur extra costs. However, suppose the key document is returned at the top of the ranking. In that case, the extra cost is unnecessary since the downstream generative module can already take advantage of the key information at spotlight positions. To avoid unnecessary costs, we introduce a preflight mechanism to predict the occurrence of "lost-in-the-middle". Such a task is challenging since the key document is unknown beforehand. Here, we employ a heuristic based on consistency across different ranking results to predict the occurrence of the issue.

We evaluated the proposed framework via both controlled experiments and integration testing. In particular, we evaluate general-purpose LLMs on answering medical QA questions that require domain knowledge in depth[1,26,38–41]. This choice of models and dataset exemplifies the scenarios where knowledge encoded from pretraining data is insufficient to answer the questions well. Our controlled experiments changed the position of key information and compared BriefContext with a regular RAG pipeline. In the integration testing, we use the ranking order from the real-world retrieval results. These experiments demonstrate that BriefContext consistently outperforms the RAG baseline by a substantial margin when the key information is placed in the middle. BriefContext also improves the model performance when the key information is placed in spotlight positions.

Furthermore, to understand how BriefContext improves the RAG pipeline, we investigate the following questions, each of which corresponds to a module in the pipeline:

(1) Can LLMs resolve conflicts correctly when the LLM context contains conflicting information? We find LLMs can correctly resolve 74.7% of cases with conflicting information in the context window. Consequently, BriefContext achieved a higher overall accuracy than the vanilla RAG.
(2) Do LLMs utilize short context more effectively than long context? Here, we prove the hypothesis that with the same key information in the context, LLMs perform better at reasoning over shorter contexts than longer ones. We controlled the number of documents in the context information and evaluated LLMs in different settings. We find that model performance decreases as the number of documents increases, even though the same key information is present in the context. This confirms that short context is utilized more effectively than long context in RAG. Furthermore, since LLMs perform better at reasoning over shorter contexts than longer ones, the problem of reasoning over long context can be divided into multiple subtasks of reasoning over short context, and the correct answer can be more easily located in one of the subtasks.
(3) How well does the preflight check predict the occurrence of "lost-in-the-middle"? We show that the preflight check can predict the issue occurrence with a recall of 92.61% but a precision of 50.18%. About 35.7% of true-negative cases can be correctly filtered by the preflight check.
(4) What is the relationship between the retrieval results and the positional attention bias? We show that positional attention bias is triggered when the key documents contain similar vocabulary to other documents in the context that do not provide supporting information to the user query.

Results

Brief context overview

Our goal is to mitigate the issue of "lost-in-the-middle", which affects the performance of RAG in QA tasks. This issue arises when the sequence of document retrieval influences the quality of the information extracted and used in generating responses.

Our proposed BriefContext consists of four modules (Fig. 1): Retrieval, Preflight check, Context Map, and Context Reduce. The Model Development section provides a detailed description. The workflow initiates when a user inputs a query. This query is converted into an encoded representation expressed as numerical vectors and used to search a knowledge base, where documents have been previously encoded into vectors using the same encoder (Retrieval module). Then, the retrieved documents are sorted using two distinct algorithms: MedCPT and BM25. It is important to note that MedCPT is also used in the primary retrieval module. By comparing the two rankings, we develop the Preflight check module to conjecture the existence of the "lost-in-the-middle" issue. If the issue is detected, the ContextMap module engages. Here, the retrieved documents are partitioned. Using partitions created in the ContextMap step, the LLMs are prompted to extract relevant information from each partition. Furthermore, the extracted responses are collected and injected into the ContextReduce module. Here, the aggregated responses undergo a summarization process to distill the most pertinent information. Finally, the summarized information is formatted into a coherent response and provided to the user as the final answer.

This workflow is designed to minimize the detrimental effects of retrieval order by reshaping how information is processed and integrated from various sources. By doing so, the BriefContext enhances the accuracy and reliability of responses in QA tasks, ensuring that users receive precise and relevant information regardless of how the initial data was retrieved.

We tested the workflow on multiple-choice questions, which allow scalable evaluation. The multiple-choice questions are all publicly available. Specifically, we chose the MIRAGE benchmark for this purpose. For a comprehensive test, we also evaluated the workflow on open-ended questions generated using publicly available education materials. The details are described in the Method section.

Can we address the issue of "lost-in-the-middle" without changing model weights?

To answer this question, we evaluated BriefContext in both controlled studies with synthetic rankings and integration testing with real-world

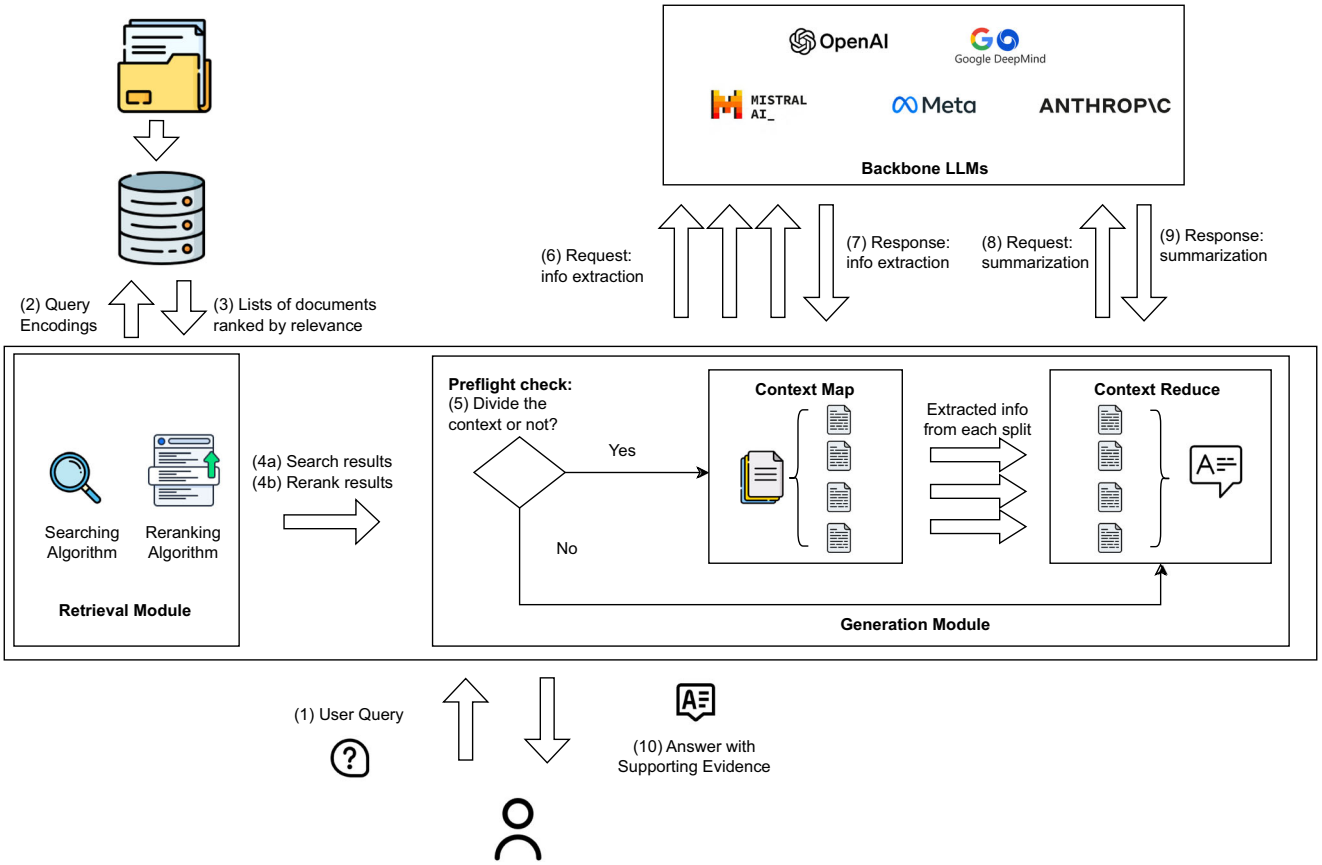

Fig. 1 | Workflow of BriefContext. In the Context Map operation (1), the retrieved documents are divided into multiple partitions to create multiple RAG subtasks. In the Context Reduce operation (2), the responses were collected from the previous step and summarized into a final response.

rankings. In the controlled study, we used all of the PubMed articles and a collection of textbooks[39] that are widely used by medical students as the knowledgebase. While a portion of the knowledge base or corpus where the dataset was derived (e.g., PubMed abstracts or textbooks) is probably included in the pre-training of LLMs, we deem the comparison remains fair, since we used the same backbone LLMs for RAG and BriefContext. We selected 20% of questions from PubMedQA[41], and MedCPT[2] as the primary search engine. The evaluation metric is accuracy, which is the ratio of correctly answered questions. As shown in Fig. 2 and Supplementary Table 2, BriefContext utilizes the external information in the middle of the context more effectively than the baseline RAG workflow (p values shown in Fig. 2 captions). Using Mixtral-7x8b[42] as the LLM backbone, the accuracy averaged over different positions was improved from 57.66 to 60.41 when top_k = 16 ($p << 0.05$). Using GPT-3.5-turbo[43] as the LLM backbone, the accuracy was improved from 54.82 to 58.11 when top_k = 8 ($p << 0.01$), and 52.12 to 58.51 when top_k = 16 ($p << 0.01$).

We used the baseline Chain-of-Thought (CoT) and RAG accuracies reported in the MIRAGE benchmark. The results of integration testing shown in Fig. 3 and Supplementary Table 3 demonstrate that BriefContext has improved the overall accuracy across different LLM backbones. With LLama2-70B-chat, the accuracy was improved from 55.81 to 66.47; with LLama3-70B-instruct, the accuracy was improved from 76.75 to 79.03; with Mixtral-7x8b, the accuracy was improved from 70.52 to 72.20; with GPT-3.5-turbo-0125, the accuracy was improved from 69.19 to 72.51. We also invited three medical experts to evaluate model responses to 48 open-ended medical questions. Out of the 48 questions, our method generates better answers than the RAG baseline for 29.2% of questions and worse answers for 12.5% of questions. For the remaining 58.3% of questions, our method and the RAG baseline produced the same responses.

Can LLMs resolve the conflicts in the retrieved external knowledge in the ContextReduce step?

In the BriefContext workflow, we divided the long text into multiple partitions. One issue is that LLM answers based on different context partitions are not always the same. We refer to such a situation as context with conflict information. It's unclear how LLMs deal with such a context. To investigate this problem, we used 20% of PubMedQA questions with synthesized rankings. The experimental setup, including the knowledge base, search engine, and backbone LLMs, is the same as the above control studies of BriefContext.

The results are shown in Fig. 4. Overall, Mixtral-7x8b resolved 171 out of 217 cases with conflicting contextual information correctly; GPT-3.5-turbo resolved 225 out of 313 cases. We also reported the win/tie/lose ratio (defined in the Method section) details in Supplementary Table 3. Overall, BriefContext consistently demonstrates a higher win rate than lose rate, which indicates the advantages of BriefContext handling context with conflict information. The advantages are more manifested in 25th, 50th, 75th percentile of positions than the others. This also highlights that the key information is under-utilized by the vanilla RAG, especially when the context contains conflicting information.

Do LLMs favor short context over long context in the ContextMap step?

To answer this, we used the same questions, knowledge base, and search engine as in the question above with synthetic rankings. We strategically placed key documents at different positions in the context (i.e., retrieved PubMed abstracts) and reported the average accuracy. We evaluated the same 4 LLM backbones using various numbers (top_k) of documents in the context. Figure 5 shows that the LLMs favor short over long context.

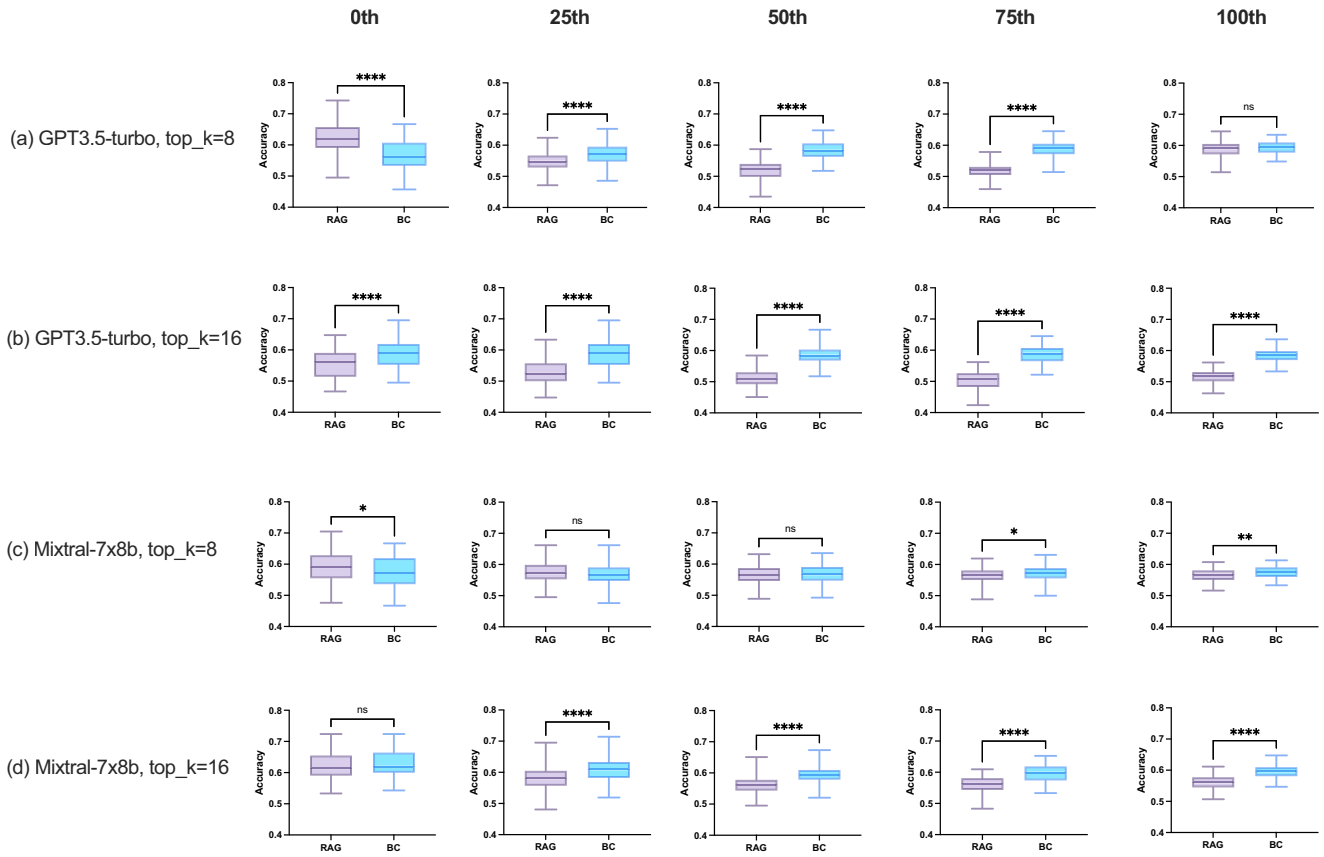

Fig. 2 | Relationship between QA accuracy and positions of key information in the LLM context. We show the average and standard deviation of accuracy of: **a**, **b** GPT-3.5-Turbo, **c**, **d** Mixtral-7x8b. The quartiles refer to the positions where the key document is located. Significance levels: *$p < 0.05$; **$p < 0.01$; ***$p < 0.001$; ****$p < 0.0001$; ns Not significant.

Can the occurrence of "lost-in-the-middle" be predicted by the Preflight check?

It typically remains unknown which documents contain the key information. It's thus unclear whether the "lost-in-the-middle" issue happens or not. To predict the occurrence of the issue, we used the consistency across different ranking results as a heuristic (see details in the Methods section). We evaluate how well the heuristic can predict the issue. We define consistency as the IoU rate between rankings from MedCPT and BM25. The threshold is set to 0.2. When the IoU is larger than 0.2, we posit that MedCPT has placed the key document at top positions, i.e., the ranking issue does not occur. To validate this hypothesis, we used precision, recall, and F1, where a true-positive is defined as an issue of "lost-in-the-middle" that happened and was successfully captured using the IoU score. The test was performed using queries from PubMedQA and BioASQ datasets, and results from PubMed were retrieved using MedCPT. The IoU heuristic achieved 50.18% precision, 92.61% recall, and 65.09% F1 (Supplementary Table 5). Based on the confusion matrix, 35.7% of true-negative cases can be correctly filtered, which avoids unnecessary procedure calls of BriefContext. Based on these numbers, we estimate the preflight check can help reduce extra overhead by up to 35%.

What is the relationship between positional attention bias and retrieval results?

Recent studies[27,28] pointed out that lost-in-middle-issue is attributed to the positional attention bias, i.e., models exhibit U-shaped attention patterns where documents at the beginning or end of the inputs receive higher attention values, regardless of their relevance. We argue that positional attention bias is related to inaccurate retrieval results that are irrelevant to the user query but contain vocabulary similar to the key documents. Recall that most modern LLM architectures employ self-attention, which calculates pair-wise inner product of embeddings as attention weights[44]. Each position is typically represented as a concatenation of position and text embedding vectors[45,46]. We hypothesize that positional attention bias is triggered only when the text embeddings of key documents are similar to other documents in the context. In other words, the positional attention bias will disappear when the key document can be associated with the query successfully and distinguished clearly from other retrieved documents in the context.

To prove this hypothesis, we randomly selected 20% of multiple-choice questions ($n = 105$) from the PubMedQA dataset. We set up two search engines to retrieve documents relevant to the questions. In the control group, we used MedCPT as the search engine and retrieved the top 16 documents from the external knowledge base using the input query. In the experimental group, we synthesize retrieval results by mixing the key documents with documents randomly selected at random from the knowledge base. The randomly selected documents were highly likely irrelevant to the input query. To manifest the lost-in-the-middle issue, we place the key document right in the middle of the LLM context for both groups. We provide the two retrieval results to downstream LLMs as contextual information and report the accuracy. Figure 6 shows that the accuracy is higher when the key documents are mixed with random documents (experimental group) as compared to relevant documents (control group), even though the key documents are placed right in the middle of the context. These results prove that positional attention bias is overpowered by text-embedding-based attention when the key information is distinguishable from other documents in the context. Furthermore, this observation highlights a limitation of search engines based on embedding representation or dense retrieval. These search engines sometimes return

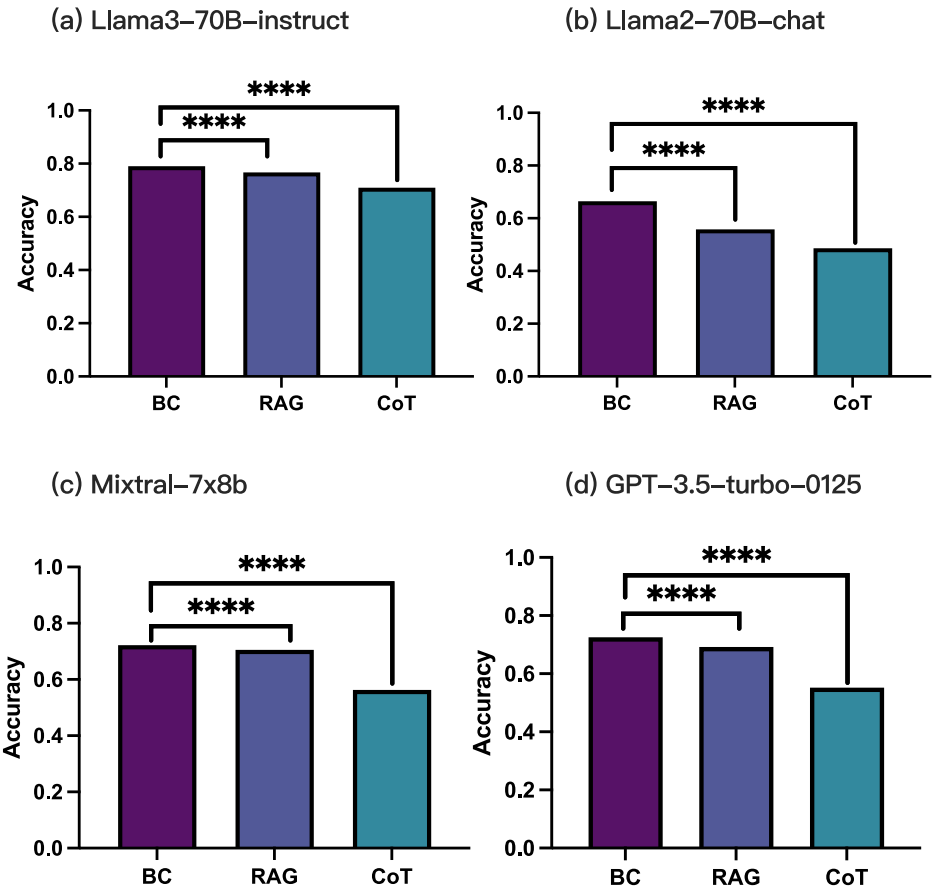

Fig. 3 | Integration testing of BriefContext with different LLM backbones. We show the accuracy of various settings with different foundation models: **a** Llama3-70B-instruct, **b** Llama2-70B-chat, **c** Mixtral-7x8b, and **d** GPT-3.5-turbo-0125. BC Brief Context. RAG Retrieval-augmented generation. CoT Chain-of-Thought. Significance levels: *$p < 0.05$; **$p < 0.01$; ***$p < 0.001$; ****$p < 0.0001$; ns Not significant.

irrelevant documents that manifest a high resemblance to the query vocabulary.

Discussion

Our experiments demonstrate that BriefContext improves the robustness regarding the order of retrieved documents in the RAG paradigm without adjusting model weights. Our proposed workflow improved accuracy on several biomedical QA datasets. This is demonstrated via both controlled studies and integration testing, as shown in Figs. 2 and 3. When conflicting information is present in the context, Mixtral-7x8b correctly resolved 78.8% of the cases with conflicting information in the context, while GPT-3.5-turbo resolved 71.8% of the cases, as shown in Supplementary Table 5. As such, the BriefContext can better utilize the key document than RAG, mainly when the context contains conflicting information. However, LLMs do not always correctly resolve the conflict information. Here, we illustrate one example where BriefContext fails, but vanilla RAG succeeds. Consider the question with ID 18507507 in PubMedQA, "*The promise of specialty pharmaceuticals: are they worth the price?*". The publication record (PMID 18507507), labeled as the key information, supports a positive answer. Other retrieved records present irrelevant information, which results in an answer with a lower level of certainty (e.g., PMID 28911475, PMID 24991326). Such retrieved records can lead to a positive answer in one partition and an uncertain answer in another. In the phase of ContextReduce, the backbone LLM favored the uncertain answer, leading to errors. Despite this, in most cases, the conflicting information can be resolved correctly.

In addition to the LLM capabilities of correctly resolving most conflicting information, we also show that key information can be better utilized in a short than a long context. To prove this, we construct various sets of context information with varying numbers of documents but the same key information. As shown in Fig. 6, the QA accuracy decreases as the number of documents is increased. By dividing a long list of documents into multiple batches, we decompose a challenging RAG task into multiple subtasks with shorter context. Resolving the "lost-in-the-middle" issue is also attributed to this division operation, which is defined as the ContextMap operation in our pipeline. In cases where the key documents are ranked at the spotlight positions, the vanilla RAG workflow can already utilize the key information. However, it's challenging to predict where the key document is ranked without knowing which document contains the key information. To combat this issue, we propose a preflight check mechanism to predict the "lost-in-the-middle" occurrence. Supplementary Table 5 shows the preflight check achieves 50.18% precision, 92.61% recall, and 65.09% F1. About 35.7% of true-negative cases can be correctly filtered by the preflight check.

Earlier studies pointed out that the issue of "lost-in-the-middle" is attributed to positional attention bias[27,28]. In this study, we show that positional attention bias only manifests when the key documents are not distinguishable from other documents in the context based on topic similarity to the query. The positional attention bias can be overpowered by the segment embeddings when the key documents are distinguishable. As shown in Fig. 6 (experimental group), the key documents can be effectively utilized even if placed right in the middle of the context. This highlights the limitations of embedding-based search engines, which mainly rely on superficial lexical similarity to perform the retrieval task without deeply understanding the relationship between user queries and the returned documents[47].

We identify the following sources of medical QA errors in the RAG paradigm. First, LLMs sometimes resolve conflicting information incorrectly. Although about 78.8% and 71.8% of conflicting information were resolved by Mixtral-7x8b and GPT-3.5, respectively, they failed to provide correct answers for the rest of the cases, resulting in wrong final answers. Second, although the RAG paradigm improves LLMs via external knowledge sources, we show that LLMs may still fail to answer questions correctly even in the oracle settings, where only key documents were provided as the

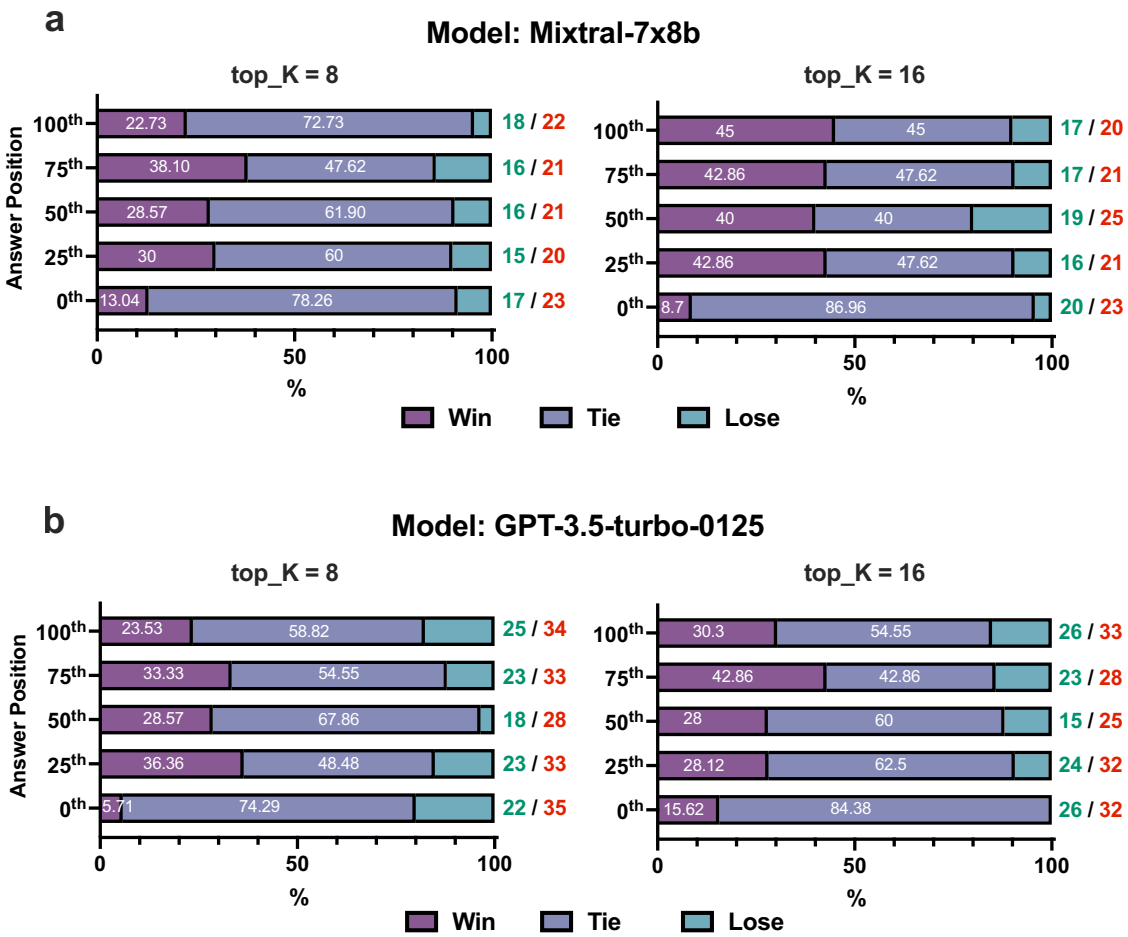

Fig. 4 | Analysis of cases with conflicting context information. Number of cases (red) with conflict information provided to LLMs and number of correctly resolved cases (green): **a** Mixtral-7x8b, **b** GPT-3.5-turbo-0125.

context. While this issue is beyond the scope of the "lost-in-the-middle", this highlights the gap between the RAG paradigm and the strict requirement for accuracy in the medical domain.

Our experiment has a few limitations. Firstly, due to the lack of open-ended questions annotated with key documents, we cannot quantitatively evaluate the impact of key document positioning on QA responses. However, we addressed this by conducting a controlled experiment using multiple-choice questions where the key document was strategically placed at various positions within the prompt context. Secondly, our choice of the off-the-shelf LLMs without any modifications presents another limitation. A future direction of this work could explore the context map-reduce paradigm with fine-tuned or task-specific LLMs. Lastly, our current focus is on QA tasks in the medical domain. In future studies, we plan to explore the application of LLMs to other tasks and QA tasks in other scientific domains. Another future direction is to incorporate the BriefContext mechanism into conversational agents to allow further validation using real-word clinical queries.

In summary, we propose BriefContext, a map-reduce approach, to effectively utilize long context in RAG workflow for answering questions in the medical domain. First, we showed that LLMs can better utilize short context than long context. Next, by dividing the long context into several subtasks, we improve the model performance on biomedical QA tasks without adjusting model weights. To avoid unnecessary extra costs on LLMs service, we then introduced a preflight check mechanism to prognose the ranking issue without knowing which document contains key information. We show LLMs can correctly resolve 74.7% of cases with conflicting information in the context window. BriefContext takes advantage of this capability of LLMs and shorter context, which explains how BriefContext improves biomedical QA accuracy in RAG workflow. Lastly, we discussed when positional attention bias is triggered. We hope this assists future research on the root cause of the positional attention bias.

While our proposed BriefContext framework was evaluated only within the biomedical question-answering in this study, it shows promise for generalizing to tasks that require effective processing of long-context data, such as extracting pertinent data from lengthy, duplicative electronic health records, legal document analysis, historical research, or technical report summarization. Future studies could explore these applications to evaluate the generalizability and adaptability of BriefContext in addressing diverse and complex information retrieval challenges.

Methods

We describe the methods in detail in four main sections, aligning with the study aims and the Results section.

To develop the model and ensure its scalable evaluation, we used multiple-choice questions, where the correctness of model outputs can be determined without necessitating further expert feedback. We chose the MIRAGE[26] benchmark for this purpose, which consists of three medical examination QA subsets (MMLU-Med[48], MedQA-US[39], and MedMCQA[49]) and two biomedical research QA subsets (PubMedQA[41] and BioASQ-Y/N[40]) (Supplementary Table 6).

Given that our goal is to improve RAG pipelines, we specifically used two biomedical subsets (PubMedQA and BioASQ-Y/N), due to their reliance on external knowledge databases that can augment the capabilities of LLMs. Furthermore, to maintain a diversity of question types, we used MedMCQA, the largest medical examination QA dataset. We particularly focus on PubMedQA and BioASQ-Y/N, which contain explicit

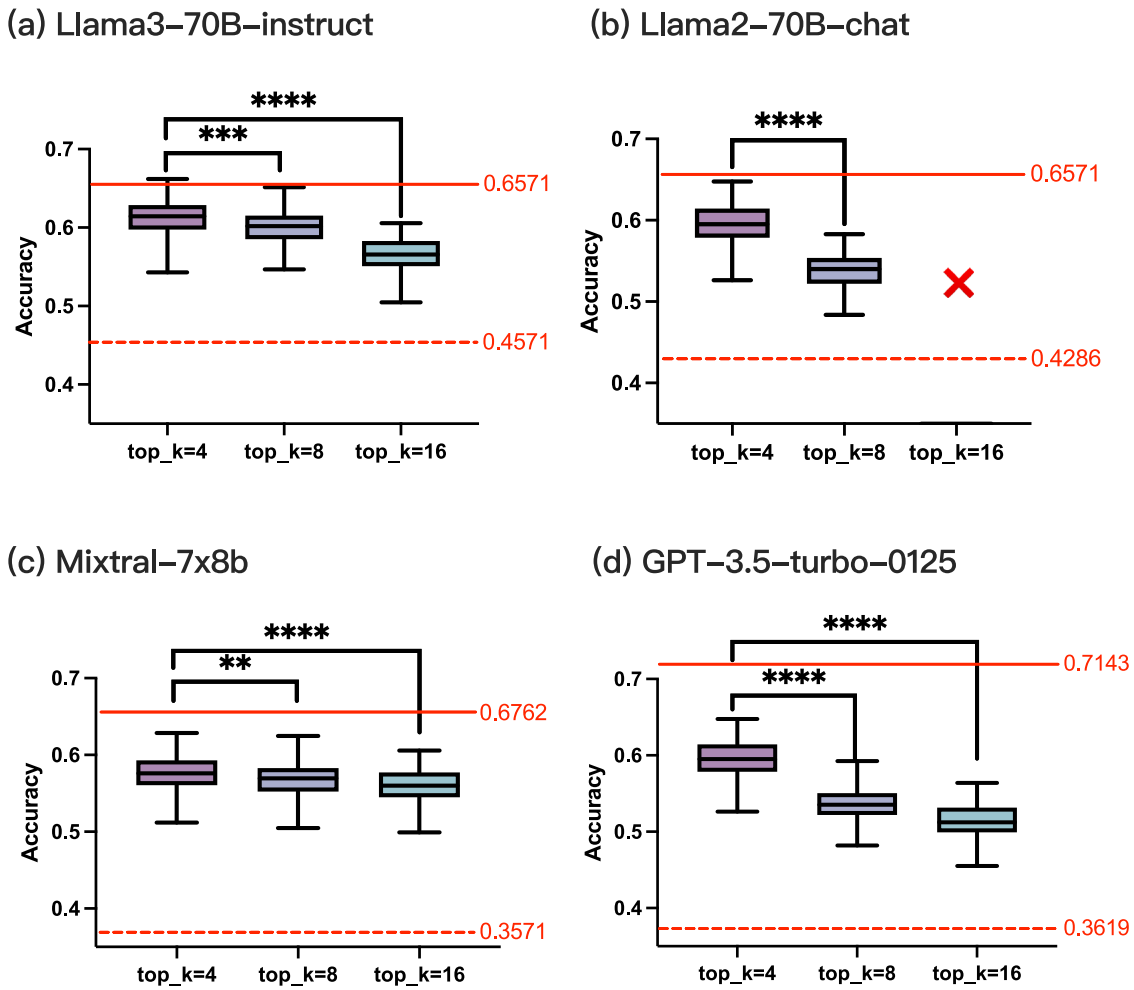

Fig. 5 | Medical QA accuracy of LLMs with various numbers of documents as context information. We show the mean and standard deviation of accuracy with different number of documents in the context window. The top solid line shows the performance in the Oracle settings. The bottom dotted line shows the performance of CoT. With the same key document in the context, the accuracy decreases as the number of documents increases. **a** Llama3-70B-instruct, **b** Llama2-70B-chat, **c** Mixtral-7x8b, and **d** GPT-3.5-turbo-0125. BC Brief Context. RAG Retrieval-augmented generation. CoT Chain-of-Thought. Significance levels: *$p < 0.05$; **$p < 0.01$; ***$p < 0.001$; ****$p < 0.0001$; ns Not significant.

specifications of the key documents. This information allows us to perform a deep analysis the relationship between the key document position and the pipeline accuracy.

In the real-world practice of medical QA, questions always arise without predefined options, reflecting the open-ended nature of real-world scenarios. As such, we present MedQ, a dataset comprising 48 open-ended questions. We created these questions using StatPearls[50], a source that summarizes up-to-date medical knowledge and practice across various specialties. In particular, we selected articles focusing on neurology, endocrinology, and dermatology.

To formulate the questions, we prompt GPT-4 to generate pairs of PICO (participant, intervention/comparison, and outcomes) questions and answers. The generated QA pairs were then reviewed by three specialties (dermatology, neurology, and endocrinology) to ensure their accuracy and relevance.

Following the practice of this benchmark work[26], we built a knowledge base with components: (1) The entire collection of abstracts indexed in PubMed, and (2) a set of 18 medical textbooks[39] (available at https://github.com/jind11/MedQA) that are widely used by medical students and serve as preparation materials for the USMLE exams (Supplementary Table 7).

Given a large collection of documents $\mathbb{D}$, the main goal of the retrieval module is to select a subset of documents $D_r = \{d_1, d_2, \ldots d_k\} \subset \mathbb{D}$ relevant to the user query Q, where k is the number of retrieved documents. To perform an effective and efficient retrieval, we first encode each document d_i and the query Q into numerical vectors of the same fixed dimension, denoted as $embed(d_i)$ and $embed(Q)$, respectively.

The collection is then sorted by the relevance to the query. We denote the resultant ranking as $R_{LLM} = [r_1{}^{LLM}, r_2{}^{LLM}, \ldots, r_k{}^{LLM}]$. where the relevance of a document d_i to a query is determined by the inner product of the two embedding vectors $r_i{}^{LLM} = embed(d_i)^\top embed(Q)$. Based on the ranking results, we discuss two possible outcomes. First, when the key information is ranked at the top positions, the generative module can take advantage of the retrieved information. In this case, there is no need to include too many articles in the context. Several results ranked at the top provide enough information to answer the question. Second, when the key information is ranked beyond the spotlight positions, the key information is probably to be neglected.

To combat ranking related issues, a common approach is to employ hybrid rankings, which ensemble several ranking results into a new order using reciprocal ranking fusion (RRF). While RRF demonstrated advantages in end-to-end RAG evaluation, there is no guarantee that documents with key information will always be placed at top positions in the hybrid ranking results. The hybrid ranking results still leaves the "lost-in-the-middle" issue unresolved. It's also unrealistic to expect any retrieval system to always place the documents of interest at the very first position.

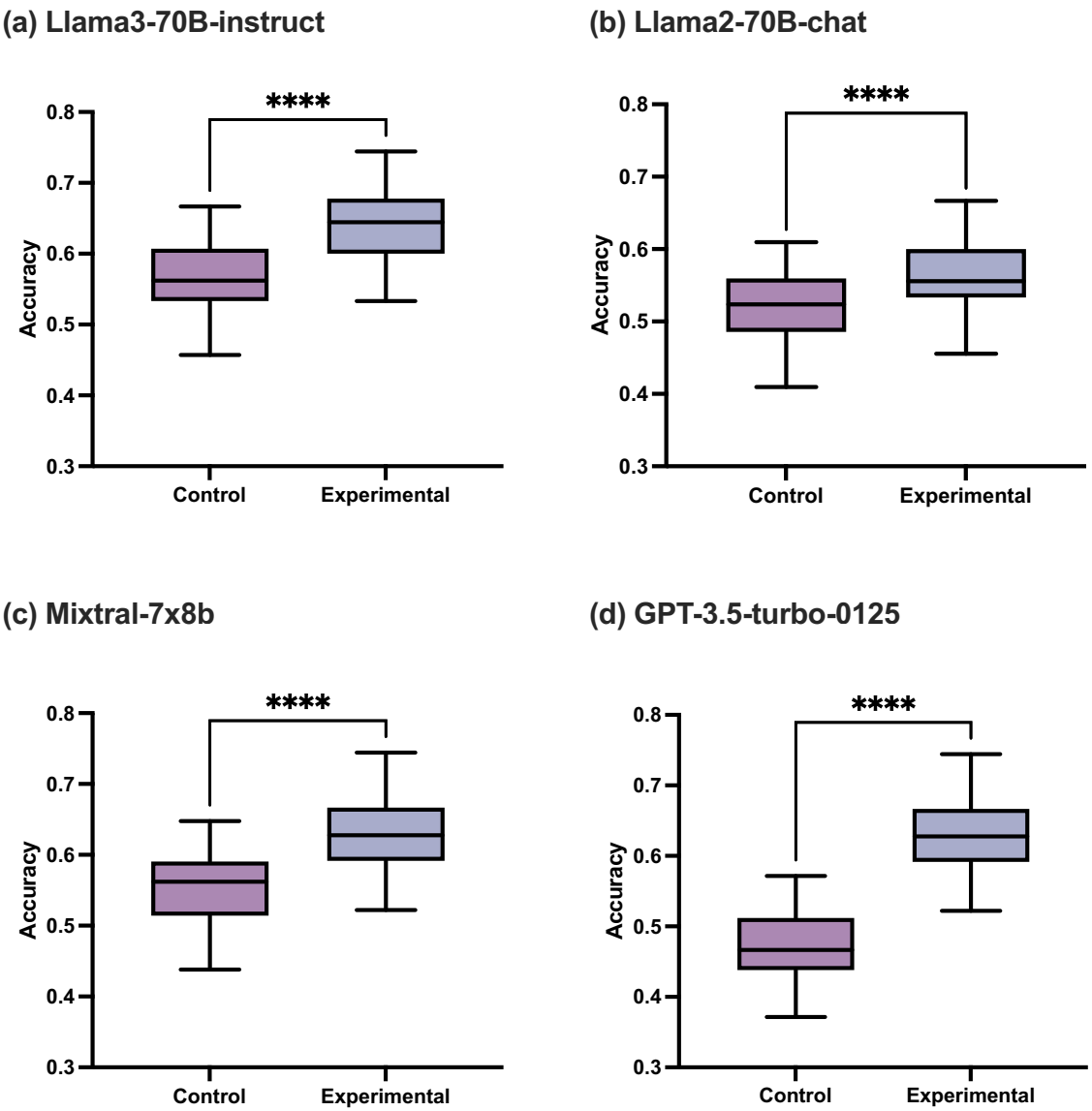

Fig. 6 | Relationship between QA accuracy and different context information. We show the average mean and standard deviation of accuracy with the real retrieval and controlled settings as the context. In the Control group, all documents come from results returned by MedCPT. In the experimental group, the context consists of key documents and others selected at random from the knowledge base. **a** Llama3-70B-instruct, **b** Llama2-70B-chat, **c** Mixtral-7x8b, and **d** GPT-3.5-turbo-0125ontext. RAG retrieval-augmented generation. CoT chain-of-thought. Significance levels: $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$; $^{****}p < 0.0001$; ns Not significant.

Inspired by the hybrid ranking algorithms, we use the consistency among different retrieval systems to conject the occurrence of ranking issues without knowing which documents contain key information. In particular, we calculate the intersection-over-union (IoU) rate between the top n results. In addition to the retrieval system based on dense representation of documents, we use another ranking algorithm, BM25, to rerank the documents in R_{LLM}. The new ranking is denoted as $R_{BM25} = [r_1^{BM}, r_2^{BM}, ..., r_k^{BM}]$. Next, we conduct a preflight check to determine whether to invoke the Brief-Context subroutine in the RAG pipeline. The preflight check is formally defined as an indicator function,

$$1(R_{LLM}, R_{BM25}, n) = \begin{cases} 1 & \text{if } \frac{R_{LLM}[:n] \cap R_{BM}[:n]}{R_{LLM}[:n] \cup R_{BM}[:n]} > \delta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The choice of the threshold is crucial in balancing the trade-off between precision and recall. While it's possible to further optimize the precision and overall F1 by adjusting the threshold, we chose a value that ensures high recall. This decision is based on the fact that false positive errors result in extra cost, while false negative errors could leave the errors in vanilla RAG unaddressed. To better demonstrate the effectiveness of our methods, we prioritize achieving high recall over high precision.

The ContextMap operation divides D_r into a partition $P(D_r)$ (i.e., the sets in P are subsets of D_r, and the elements of P are mutually exclusive) and converts each subset as a prompt. Here, each subset has the same number of documents, denoted as $D_r^s \in P$. The output is a list of prompts with the same instruction and user query, as outlined in Supplementary Note 1. Consider a partition of $D_r = \{d_1, d_2, \ldots d_8\}$ as $D_r^1 = \{d_1, d_2, d_3, d_4\}$ and $D_r^2 = \{d_5, d_6, d_7, d_8\}$, the resultant prompts are "*{instruction} {query} [doc 1]*d_1 *[doc 2]* d_2… " and "*{instruction} {query} [doc 1]*d_5 *[doc 2]* d_6… ". The only difference between the prompts is the contextualized documents. It has been pointed out that decoder-only models cannot attend to query tokens if the query is only placed behind the contextual information, since decoder-only models only attend to prior tokens by each timestamp. To combat this effect, we adopt query-aware contextualization, where a prompt consists of instruction, context information, and the user query placed before the context. Since not all documents in D_r are necessarily related to the query Q, we instruct the model to either extract the relevant information

or truthfully report no detection of any relevant information. The operation of ContextMap can be processed in parallel via multi-threading, where each thread formats a prompt. This batch processing is straightforward to implement since the prompt formatting subroutine only requires read access to the context.

After the context mapping, we next query the backbone LLM to extract relevant information from the context and answer the user query, as outlined in Supplementary Note 2. The relevant information is autoregressively sampled from the probability distribution over the model vocabulary conditioned on the instruction, query, and provided context:

$$y_t^{info} \sim p_\theta(Q, D_{rs}^S, I_e, y_{0:t-1}^{info}), \quad (2)$$

where we denote the model weights as θ, extraction instruction I_e, query Q, shard of context D_{rs}^S, and y_t^{info} the sampled information. The invocations to extraction can also be streamlined in parallel. The extracted information is then used to generate a summarization prompt, where we provide instructions I_s to ignore empty information. The final answer is also directly sampled from the probability distribution over the model vocabulary conditioned on the summarization instruction, extracted information, and query:

$$y_t^{answer} \sim p_\theta(Q, y^{info}, I_s, y_{0:t-1}^{answer}). \quad (3)$$

As in a typical map-reduce workflow, the long context of relevant documents is first divided and dispatched to worker LLMs to create requests for extracting relevant information. After all the worker LLMs finish their processing jobs, they return to the LLM allocator to aggregate the individual results.

Below, we discuss the extra cost incurred by invoking the BriefContext subroutine. Here, we use the pricing model of most proprietary LLMs, e.g., GPTs, where users are charged by the number of input and output tokens. We denote the number of tokens in the prompt instruction and context as N_{ins} and N_{con}, and the maximum number of output tokens as N_{out}, respectively. The prices of input and output per token are denoted as p_{input} and p_{output}. The context is divided into M partitions. The cost of vanilla RAG is

$$O(N_{con} \cdot p_{input} + N_{ins} \cdot p_{input} + N_{out} \cdot p_{out}) \quad (4)$$

while the cost of BriefContext invocation is

$$O(N_{con} \cdot p_{input} + M \cdot N_{ins} \cdot p_{input} + (M+1) \cdot N_{out} \cdot p_{out}) \quad (5)$$

Since the lengths of instruction and output are much shorter than the context information, the extra cost incurred by BriefContext invocations is not significant in scale.

In our cost analysis, we adopted the big-O notation, proving that BriefContext and vanilla RAG are at the same level in terms of theoretical complexity. However, in real-world scenarios, constant factors do play a role. While these extra tokens do not impact the big-O analysis, they would still increase the actual costs. It's challenging to accurately quantify the percentage increase in cost since this varies by the specific prompt, retrieved documents, and batch size in BriefContext. In case the context is as short as a prompt instruction, the lost-in-the-middle issue would not appear, thus not requiring the BriefContext procedure.

Another factor to consider is the occurrence of "lost-in-the-middle" issues, which can vary by the queries, corpus, and choice of retrieval models. To help understand the frequency of these issues, we reported the average number of tokens per request, with 8 publication records retrieved for each query. In BriefContext, the average numbers of input and output tokens per request are 5496.5 and 247.5, respectively. In vanilla RAG, these numbers are 3066.0 and 183.0.

Can we address the issue of "lost-in-the-middle" without changing model weights?

To answer this question, we evaluated BriefContext in both controlled studies with synthetic rankings and integration testing with real-world rankings. In the controlled study, we used the same experimental setup as the above question, i.e., the knowledge base of PubMed articles and textbooks, the 20% subset of questions from PubMedQA, and MedCPT as the primary search engine. The evaluation metric is accuracy. We synthesized rankings by placing key information at different positions, including 0th, 25th, 50th, 75th, and 100th percentile of positions in the context. We used Mixtral-7x8B and GPT-3.5-turbo as LLM backbones since these two models benefit more from retrieval augmentation than others (Fig. 2). We compared the BriefContext with the vanilla RAG workflow using the same backbone LLM and external knowledge as the context in the prompts.

In the first integration testing, we used all questions from MedMCQA[49], PubMedQA[41], and BioASQ-Y/N[40] from the MIRAGE[26] benchmark dataset. The evaluation metric is accuracy. We selected LLama2-70B-chat[51], LLama3-70B-instruct[52], Mixtral-7x8b[42], and GPT-3.5-turbo[43] as backbone LLMs, all of which have been used in the published benchmark results[26]. We used the baseline closed-book (CoT) and RAG accuracies that were reported in the MIRAGE benchmark results[26]. We used the same knowledge base as in the controlled studies. The knowledge base is a subset of the corpus that was used in the MIRAGE benchmark results reported by Xiong et al.[26]. Our knowledge base thus contains no extra information as compared to theirs, which makes a fair comparison between BriefContext and RAG. In BriefContext, we used MedCPT as the search engine. The order of retrieved documents by MedCPT was kept the same when the prompt context was constructed. The top_k is set to 16. In the second integration testing, we invited three medical experts to help curate 48 open-ended question-answer pairs from their specialty domain and compare our method with the RAG baseline.

Can LLMs resolve the conflicts in the retrieved external knowledge in the ContextReduce step?

To investigate this problem, we used 20% of PubMedQA questions with synthesized rankings. The experimental setup, including the knowledge base, search engine, and the backbone LLM, is the same as the above experiments. We define the occurrence of conflict information as an event in which LLMs return inconsistent answers given different context partitions of the same query results. We further define that the conflict is correctly resolved if the final answer is correct. We report the number of cases with conflict information and how many cases were correctly resolved by our proposed workflow. We also compare BriefContext with the vanilla RAG, which has the same backbone LLM in these cases. The comparison results consist of three possible outcomes: (1) our method wins the comparison if it resolves the conflict information correctly while the RAG baseline answers the question incorrectly; (2) the lose outcome is defined similarly; or (3) the outcome is a tie when both BriefContext and RAG answer the question either correctly or incorrectly.

Do LLMs favor short context over long context in the ContextMap step?

To answer this, we used the same questions, knowledge base, and search engine as in the question above with synthetic rankings. We strategically placed key documents at the 0th, 25th, 50th, 75th, and 100th percentile of the positions in the context (i.e., retrieved PubMed abstracts) and calculated the accuracy averaged over the five positions.

Can the occurrence of "lost-in-the-middle" be predicted by the Preflight check?

To predict the occurrence of the issue, we used the consistency across different ranking results as a heuristic. We evaluate how well the heuristic can predict the issue. In this experiment, we selected all questions from PubMedQA and BioASQ, where the answers were also annotated with the PMID of articles that contained the key information. The issue occurrence is

defined as an event where the key document is ranked beyond the top N positions. The threshold N is set to 3 since model performance drops significantly when N becomes larger than 3, according to earlier studies[24,27,28]. We used the same knowledge base as in the above questions. We used MedCPT as the primary search engine and BM25 as the secondary search engine to rerank the retrieval results from MedCPT. We define consistency as the IoU rate between rankings from MedCPT and BM25. The threshold is set to 0.2.

What is the relationship between positional attention bias and retrieval results?

In our study, we decouple the impact of segment embeddings on attention weights from the impact of positional embeddings. Recall that Transformer architecture adopts the self-attention mechanism, where the weight is calculated as an inner-product between each pair of embeddings[44]. Each embedding consists of positional, token, and segment embeddings, which encode position and semantics, respectively[20,45]. We randomly selected 20% of multiple-choice questions ($n = 105$) from the PubMedQA dataset. Each question is a multiple-choice question, and the evaluation metric is accuracy. The knowledge base consists of two components. One is all of the abstracts indexed at PubMed (https://pubmed.ncbi.nlm.nih.gov/), and the other is a collection of 18 textbooks[39] that medical students widely use for preparing USMLE. Scientific literature has certain limitations, such as publication bias and lack of generalizability across different population groups. Another potential supplementary information source is real-world clinical data, such as electronic health records. However, due to the sensitive nature, retrieval augmented LLMs based on clinical data is out of the scope of this study. We used MedCPT[2] as the search engine to obtain relevant information from the knowledge base. MedCPT was specifically pretrained on biomedical literature using user click information[2].

Data availability

The results are provided within the supplementary information files.

Code availability

The code is available at https://github.com/ebmlab/BriefContext under the MIT license.

Received: 13 September 2024; Accepted: 19 April 2025;
Published online: 02 May 2025

References

1. Singhal, K. et al. Large language models encode clinical knowledge. *Nature* **620**, 172–180 (2023).
2. Jin, Q. et al. MedCPT: contrastive pre-trained transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval. *Bioinformatics* **39**, btad651 (2023).
3. Haupt, C. E. & Marks, M. AI-Generated Medical Advice—GPT and Beyond. *JAMA* **329**, 1349–1350 (2023).
4. Peng, Y., Rousseau, J. F., Shortliffe, E. H. & Weng, C. AI-generated text may have a role in evidence-based medicine. *Nat. Med.* **29**, 1593–1594 (2023).
5. Zhang, G. et al. A span-based model for extracting overlapping PICO entities from randomized controlled trial publications. *J. Am. Med. Inform. Assoc.* **31**, 1163–1171 (2024).
6. Idnay, B. et al. Mini-mental status examination phenotyping for Alzheimer's disease patients using both structured and narrative electronic health record features. *J. Am. Med. Inform. Assoc*. https://doi.org/10.1093/jamia/ocae274 (2024).
7. Jin, Q. et al. Demystifying large language models for medicine: a primer. *arXiv [cs.AI]* (2024).
8. Spotnitz, M. et al. A survey of clinicians' views of the utility of large language models. *Appl. Clin. Inform.* **15**, 306–312 (2024).
9. Zelin, C., Weng, C., Jeanne, M., Zhang, G. & Weng, C. Rare disease diagnosis using knowledge guided retrieval augmentation for ChatGPT. *J. Biomed. Inform.* **157**, 104702 (2024).
10. Tian, S. et al. Opportunities and challenges for ChatGPT and large language models in biomedicine and health. *Brief. Bioinform*. **25**, bbad493 (2023).
11. Zhang, G. et al. Closing the gap between open source and commercial large language models for medical evidence summarization. *NPJ Digit. Med.* **7**, 239 (2024).
12. Park, J. et al. Criteria2Query 3.0: leveraging generative large language models for clinical trial eligibility query generation. *J. Biomed. Inform.* **154**, 104649 (2024).
13. Cao, M., Dong, Y., Wu, J. & Cheung, J. C. K. Factual error correction for abstractive summarization models. In *Proc. 2020 Conf. Empirical Methods in Natural Language Processing (EMNLP)* 6251–6258 (Association for Computational Linguistics, 2020). https://doi.org/10.18653/v1/2020.emnlp-main.506.
14. Chen, J., Lin, H., Han, X. & Sun, L. Benchmarking large language models in retrieval-augmented generation. *AAAI* **38**, 17754–17762 (2024).
15. Raunak, V., Menezes, A. & Junczys-Dowmunt, M. The curious case of hallucinations in neural machine translation. In *Proc. 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* 1172–1183 (Association for Computational Linguistics, 2021). https://doi.org/10.18653/v1/2021.naacl-main.92.
16. Ji, Z. et al. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **55**, 1–38 (2023).
17. Zhang, G. et al. Leveraging generative AI for clinical evidence synthesis needs to ensure trustworthiness. *J. Biomed. Inform.* **153**, 104640 (2024).
18. Guu, K., Lee, K., Tung, Z., Pasupat, P. & Chang, M. Retrieval augmented language model pre-training. In *Proc. 37th International Conference on Machine Learning* (eds. Iii, H. D. & Singh, A.) 3929–3938 (PMLR, 2020).
19. Lewis, P., Perez, E., Piktus, A. & Petroni, F. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inform. Process. Syst.* **33**, 9459–9474 (2020).
20. Borgeaud, S. et al. Improving language models by retrieving from trillions of tokens. In *Proc. 39th International Conference on Machine Learning* (eds. Chaudhuri, K. et al.)2206–2240 (PMLR, 2022).
21. Izacard, G. & Grave, E. Leveraging passage retrieval with generative models for open domain question answering. In *Proc. 16th Conf. of the European Chapter of the Association for Computational Linguistics (EACL)* 874–880 (Association for Computational Linguistics, 2021). https://doi.org/10.18653/v1/2021.eacl-main.74.
22. Ding, Y. et al. A Survey on RAG Meets LLMs: towards retrieval-augmented large language models. *arXiv [cs.CL]* (2024).
23. Li T, Zhang G, Do QD, Yue X, Chen W. Long-context LLMs struggle with long in-context learning. *arXiv* [csCL]. 2024.
24. Liu, N. F. et al. Lost in the middle: how language models use long contexts. *Trans. Assoc. Comput. Linguist.* **12**, 157–173 (2023).
25. Jiang, H. et al. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proc. 62nd Annu. Meet. Assoc. Comput. Linguistics (ACL)* 1658–1677 (Association for Computational Linguistics, 2024). https://doi.org/10.18653/v1/2024.acl-long.91.
26. Xiong, G., Jin, Q., Lu, Z. & Zhang, A. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024* 6233–6251 (2024). https://doi.org/10.18653/V1/2024.FINDINGS-ACL.372.
27. He, J. et al. Never lost in the middle: improving large language models via attention strengthening question answering. *arXiv [cs.CL]* (2023).
28. Hsieh, C.-Y. et al. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics (ACL)* 14982–14995 (Association for Computational Linguistics, 2024). https://doi.org/10.18653/v1/2024.findings-acl.890.

29. Wu, T., Zhao, Y. & Zheng, Z. An efficient recipe for long context extension via middle-focused positional encoding. In *Adv. Neural Inf. Process. Syst. 38* (NeurIPS, 2024).
30. Kemker, R., McClure, M., Abitino, A., Hayes, T. & Kanan, C. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32, No. 1, 3390–3398 (2018).
31. Kirkpatrick, J. et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA.* **114**, 3521–3526 (2017).
32. Dean, J. & Ghemawat, S. MapReduce. *Commun. ACM* **53**, 72–77 (2010).
33. Zhang, H., Li, P., Meng, F., Fan, W. & Xue, Z. MapReduce-based distributed tensor clustering algorithm. *Neural Comput. Appl.* **35**, 24633–24649 (2023).
34. Bergui, M., Hourri, S., Najah, S. & Nikolov, N. S. Predictive modelling of MapReduce job performance in cloud environments using machine learning techniques. *J. Big Data* **11**, 98 (2024).
35. Senthamil Selvi, R., Sankari, V., Ramya, N. & Selvi, M. Ensemble model for stock price forecasting: MapReduce framework for big data handling: an optimal trained hybrid model for classification. *J. Circuits Syst. Comput*. **33**, 2450202 (2024).
36. Mv, K. et al. Survey on MapReduce scheduler algorithms in Hadoop framework. *Int. J. Innov. Res. Inf. Secur.* **10**, 314–319 (2024).
37. Luo, Y. & Li, J. Face Image Encryption Using Fuzzy K2DPCA and Chaotic MapReduce. *Tehnički vjesnik* **31**, 1143–1153 (2024).
38. Liévin, V., Hother, C. E., Motzfeldt, A. G. & Winther, O. Can large language models reason about medical questions? *Patterns* **5**, 100943 (2024).
39. Jin, D. et al. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *NATO Adv. Sci. Inst. Ser. E Appl. Sci.* **11**, 6421 (2021).
40. Tsatsaronis, G. et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics* **16**, 138 (2015).
41. Jin, Q., Dhingra, B., Liu, Z., Cohen, W. & Lu, X. PubMedQA: a dataset for biomedical research question answering. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* 2567–2577 (Association for Computational Linguistics, 2019).
42. Jiang, A. Q. et al. Mixtral of Experts. *arXiv [cs.LG]* (2024).
43. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
44. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017).
45. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* 4171–4186 (Association for Computational Linguistics, 2019). https://doi.org/10.18653/v1/n19-1423.
46. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training. OpenAI. *Preprint*, pp.1–12, (2018).
47. Steck, H., Ekanadham, C. & Kallus, N. Is cosine-similarity of embeddings really about similarity? in *Companion Proceedings of the ACM on Web Conference 2024*. https://doi.org/10.1145/3589335.3651526 (ACM, 2024).
48. Hendrycks, D. et al. Measuring massive multitask language understanding. In *Proc. 9th Int. Conf. Learn. Represent. (ICLR)* (OpenReview.net, 2021).
49. Pal, A., Umapathi, L. K. & Sankarasubbu, M. MedMCQA: a large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proc. Conference on Health, Inference, and Learning* (eds. Flores, G., Chen, G. H., Pollard, T., Ho, J. C. & Naumann, T.) 248–260 (PMLR, 2022).
50. StatPearls Publishing. (n.d.). *StatPearls*. Retrieved September 5th, 2024, from https://www.ncbi.nlm.nih.gov/books/NBK430685/.
51. Touvron, H. et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv [cs.CL]* (2023).
52. Introducing Meta Llama 3: The most capable openly available LLM to date. *Meta AI* https://ai.meta.com/blog/meta-llama-3/.

Acknowledgements

This project was sponsored by the National Library of Medicine grant R01LM009886, R01LM014344, and R01LM014573, the National Center for Advancing Clinical and Translational Science awards UL1TR001873 and UL1TR002384. Q.J. and Z.L are supported by the NIH Intramural Research Program, National Library of Medicine. We also want to express our gratitude to Amazon Web Services (AWS) for providing the computational resources used in our research.

Author contributions

Study concepts/study design, G.Z., C.W., and Y.P.; manuscript drafting or manuscript revision for important intellectual content, G.Z., Z.X., Q.J., F.C., Y.F., Y.L., J.F.R., Z.X., Z.L., C.W., and Y.P.; approval of final version of the submitted manuscript, G.Z., Z.X., Q.J., F.C., Y.F., Y.L., J.F.R., Z.X., Z.L., C.W., and Y.P.; agrees to ensure any questions related to the work are appropriately resolved, G.Z., Z.X., Q.J., F.C., Y.F., Y.L., J.F.R., Z.X., Z.L., C.W., and Y.P.; literature research, G.Z. and Y.P.; experimental studies, G.Z., Z.X., Q.J., F.C., and Y.F.; human evaluation, Y.L., J.F.R., and Z.X.; data interpretation and statistical analysis, G.Z. and Y.P.; and manuscript editing, G.Z., Z.X., Q.J., F.C., Y.F., Y.L., J.F.R., Z.X., Z.L., C.W., Y.P.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41746-025-01651-w.

Correspondence and requests for materials should be addressed to Chunhua Weng or Yifan Peng.