# CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models

YUANJIE LYU, University of Science and Technology of China, Hefei, China
ZHIYU LI, Institute for Advanced Algorithms Research (Shanghai), Shanghai, China
SIMIN NIU, Renmin University of China, Beijing, China
FEIYU XIONG, BO TANG, WENJIN WANG, and HAO WU, Institute for Advanced Algorithms Research (Shanghai), Shanghai, China
HUANYONG LIU, 360 AI Research Institute, Beijing, China
TONG XU and ENHONG CHEN, University of Science and Technology of China, Hefei, China

Retrieval-augmented generation (RAG) is a technique that enhances the capabilities of large language models (LLMs) by incorporating external knowledge sources. This method addresses common LLM limitations, including outdated information and the tendency to produce inaccurate "hallucinated" content. However, evaluating RAG systems is a challenge. Most benchmarks focus primarily on question-answering applications, neglecting other potential scenarios where RAG could be beneficial. Accordingly, in the experiments, these benchmarks often assess only the LLM components of the RAG pipeline or the retriever in knowledge-intensive scenarios, overlooking the impact of external knowledge base construction and the retrieval component on the entire RAG pipeline in non-knowledge-intensive scenarios. To address these issues, this article constructs a large-scale and more comprehensive benchmark and evaluates all the components of RAG systems in various RAG application scenarios. Specifically, we refer to the CRUD actions that describe interactions between users and knowledge bases and also categorize the range of RAG applications into four distinct types—create, read, update, and delete (CRUD). "Create" refers to scenarios requiring the generation of original, varied content. "Read" involves responding to intricate questions in knowledge-intensive situations. "Update" focuses on revising and rectifying inaccuracies or inconsistencies in pre-existing texts. "Delete" pertains to the task of summarizing extensive texts into more concise forms. For each of these CRUD categories, we have developed different datasets to evaluate the performance of RAG systems. We also analyze the effects of various components of the RAG system, such as the retriever, context length, knowledge base construction, and LLM. Finally, we provide useful insights for optimizing the RAG technology for different scenarios. The source code is available at GitHub: https://github.com/IAAR-Shanghai/CRUD_RAG.

Yuanjie Lyu and Zhiyu Li contributed equally to this research.

This work was supported in part by the grants from National Science and Technology Major Project (No. 2023ZD0121104), and National Natural Science Foundation of China (No.62222213, 62072423).

Authors' Contact Information: Yuanjie Lyu, University of Science and Technology of China, Hefei, China; e-mail: s1583050085@gmail.com; Zhiyu Li, Institute for Advanced Algorithms Research (Shanghai), Shanghai, China; e-mail: lizy@iaar.ac.cn; Simin Niu, Renmin University of China, Beijing, China; e-mail: niusimin@ruc.edu.cn; Feiyu Xiong, Institute for Advanced Algorithms Research (Shanghai), China; e-mail: xiongfy@iaar.ac.cn; Bo Tang, Institute for Advanced Algorithms Research (Shanghai), China; e-mail: tangb@iaar.ac.cn; Wenjin Wang, Institute for Advanced Algorithms Research (Shanghai), China; e-mail: wangwj@iaar.ac.cn; Hao Wu, Institute for Advanced Algorithms Research (Shanghai), China; e-mail: wuh@iaar.ac.cn; Huanyong Liu, 360 AI Research Institute, Beijing, China; e-mail: liuhuanyong@360.cn; Tong Xu (corresponding author), University of Science and Technology of China, Hefei, China; e-mail: tongxu@ustc.edu.cn; Enhong Chen, University of Science and Technology of China, Hefei, China; e-mail: cheneh@ustc.edu.cn.



ACM 1558-2868/2025/1-ART41
https://doi.org/10.1145/3701228

CCS Concepts: • **Computing methodologies** → **Natural language generation**; • **Information systems** → **Information retrieval**;

Additional Key Words and Phrases: Retrieval-Augmented Generation, Large Language Models, Evaluation

**ACM Reference format:**
Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models. *ACM Trans. Inf. Syst.* 43, 2, Article 41 (January 2025), 32 pages.
https://doi.org/10.1145/3701228

## 1 Introduction

**Retrieval-augmented generation (RAG)** is an advanced technique that leverages external knowledge sources to enhance the text generation capabilities of **large language models (LLMs)**. It retrieves relevant paragraphs from a corpus based on the input and feeds them to the LLMs along with the input. With the help of external knowledge, LLMs can generate more accurate and credible responses and effectively address challenges such as outdated knowledge [19], hallucinations [3, 9, 35, 65], and lack of domain expertise [30, 46]. Therefore, RAG technology is attracting increasing attention.

Although the effectiveness of retrieval-augmented strategies has been proven through extensive practice, their implementation still requires a significant amount of tuning. The overall performance of the RAG system is affected by multiple factors, such as the retrieval model, construction of the external knowledge base, and language model. Therefore, automatic evaluation of RAG systems is crucial. Currently, there are only a few existing benchmarks for evaluating RAG performance, as creating high-quality datasets and experimenting with them entail significant costs. These benchmarks can be classified into two types: reference-required and reference-free evaluation. Reference-free evaluation frameworks, such as RAGAS [13] and ARES [44], use LLM-generated data to evaluate RAG systems on contextual relevance, faithfulness, and informativeness. These frameworks do not depend on ground truth references, but only assess the coherence of the generated text with the retrieved context. This approach may be unreliable if the retrieved external information is low-quality.

Consequently, reference-required evaluations remain the predominant method for assessing RAG systems. Existing benchmarks for reference-required evaluations, such as **Retrieval-Augmented Generation Benchmark (RGB)** [8] and **Natural Questions Benchmark (NQ)** [26], do have their limitations. First, they all rely on **question-answering (QA)** tasks to measure the performance of RAG systems. QA is not the only RAG application scenario, and an optimization strategy that works well for QA may not be generalized to other scenarios. Thus, these benchmarks may not capture the full potential of RAG systems. Second, in the experiments, current evaluations usually focus on evaluating the LLM part of the RAG pipeline, or focus on retriever performance in the knowledge-intensive scenario [40], while ignoring the retrieval methods in non-knowledge-intensive scenarios and external knowledge base construction. These components are also crucial for RAG systems. Therefore, a comprehensive evaluation of the RAG system may not be obtained using any existing benchmarks.

To evaluate the performance of RAG in different application scenarios, we need a comprehensive benchmark that covers more than just the QA task. Lewis et al. [28] argue that the core of RAG systems is their interactive way of combining LLMs with external knowledge sources. Following [25], we can group any interaction with external knowledge sources into four basic actions: create, read, update, and delete, which are also known as CRUD actions [48]. Therefore, we can use the

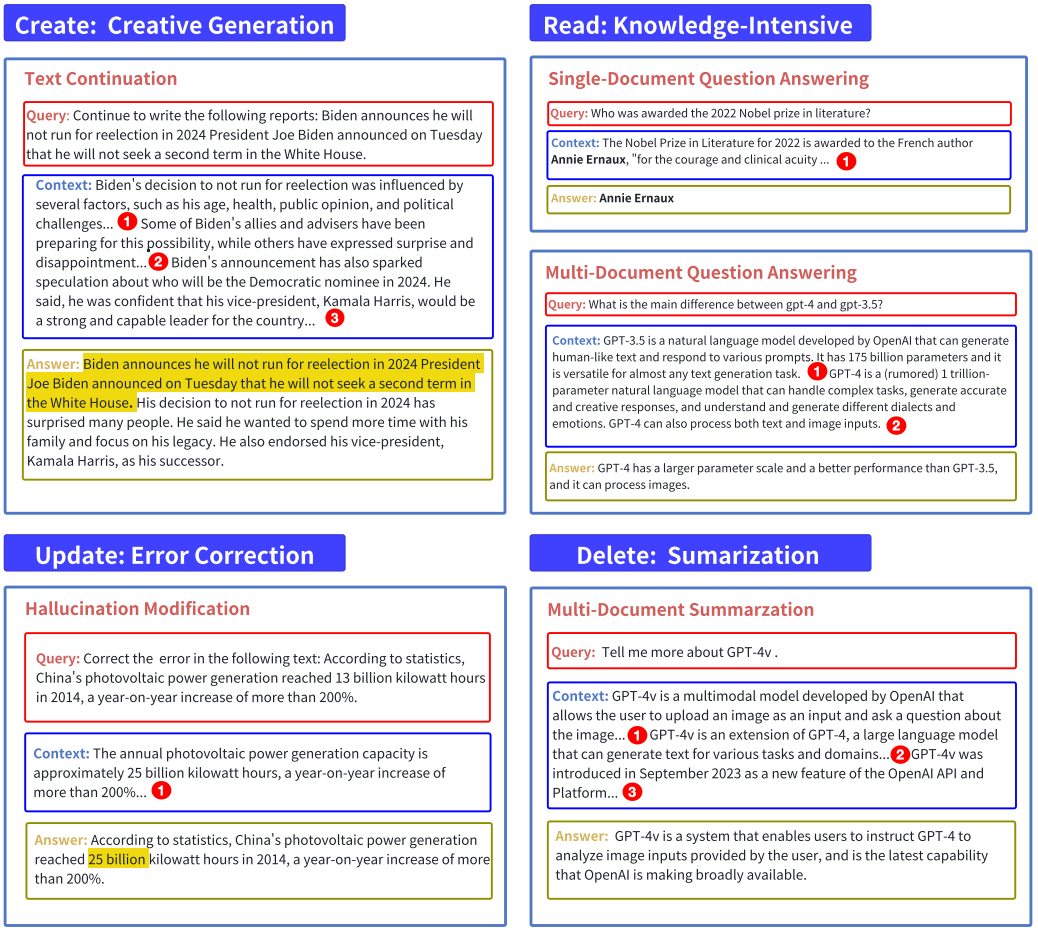


Fig. 1. We have classified the application scenarios of RAG into four primary aspects: create, read, update, and delete (CRUD). The figure provides an illustrative example for each category, showcasing the wide-ranging potential of RAG technology.

**create, read, update, and delete (CRUD)** framework to classify the RAG systems' application scenarios. As shown in Figure 1, each CRUD category demonstrates different capabilities of the RAG system:

—In "*CREATE,*" the system improves the input text by adding relevant information from external sources, making creative outputs such as poetry, stories, or code.
—In "*READ,*" the system uses external knowledge retrieval to answer questions, solve problems in QA, dialogue, and reasoning, and increase understanding of the input text.
—In "*UPDATE,*" the system fixes errors in the input text using retrieved content, correcting spelling, grammar, or factual errors to make the text better.
—In "*DELETE,*" the system simplifies the input by improving retrieval results, removing unnecessary details, and doing tasks like text summarization or simplification.

To evaluate the RAG system in these four scenarios, we introduce CRUD-RAG, a comprehensive, large-scale Chinese RAG benchmark. CRUD-RAG consists of four evaluation tasks: text continuation, QA (with single-document and multi-document questions), hallucination modification, and

open-domain multi-document summarization, which respectively correspond to the CRUD-RAG classification of RAG application scenarios. We construct CRUD-RAG by crawling the latest high-quality news data from major news websites in China, which aims to minimize the likelihood of LLMs encountering these data during training. Then, we automatically create datasets using GPT-4 based on these news data. For the multi-document summarization task, we apply a reverse construction strategy. We first generate news events and their summaries using GPT-4. Then, we use these events as keywords to search for 10 related and non-duplicate reports from the web, which we add to our retrieval database. During evaluation, the RAG system will use the retrieval database to generate summaries for the events. For the text continuation task, we split the news text into a beginning and a continuation paragraph. We then use each sentence in the continuation paragraph as a keyword to search for 10 related reports on the Web. We remove any duplicate content and add the reports to the retrieval database. For the single-document QA task, we use the RGB [8] construction method. For the multi-document QA task, we use the **chain-of-thought (CoT)** technology to help the model identify common and different aspects among documents, and then generate questions based on these aspects with increasing difficulty. For the hallucination modification task, we use the annotations in the UHGEval dataset and correct hallucinations with GPT-4. We also include the real news in UHGEval in the retrieval database.

In the experiments, we systematically evaluate the RAG system's performance on our CRUD-RAG benchmark. We also investigate various factors that affect the RAG system, such as the context length, the chunk size, the embedding model, the retrieval algorithms, and the LLM. Based on our experimental results, we provide some valuable suggestions for building effective RAG systems.

The contributions of this article are:

—*A comprehensive evaluation benchmark*: Our benchmark covers not only QA, but also CRUD of RAG applications.
—*High-quality evaluation datasets*: We constructed diverse datasets for different evaluation tasks, based on the application scenarios of RAG. These tasks include text continuation, multi-document summarization, QA, and hallucination modification.
—*Extensive experiments*: we performed extensive experiments on our benchmark, using various metrics to measure the performance of RAG systems. Based on our experiments, we offered useful guidance for future researchers and RAG system developers.

## 2 Related Work

### 2.1 RAG

LLMs excel in text generation, decision-making [52], information extraction [54], and personalized recommendations [62]; however, they also confront challenges such as outdated knowledge and the generation of hallucinatory content [6, 19, 43]. In response to these challenges, RAG, also referred to as Retrieval-Augmented Language Models, incorporates external knowledge to generate responses characterized by enhanced accuracy and realism [47]. This is particularly critical in domains that heavily depend on precision and reliability, including but not limited to the legal, medical, and financial sectors. Retrieval models have been promoting the development of language models [15, 33, 61].

Conventional RAG systems adhere to a standardized workflow encompassing indexing, retrieval, and generation phases [28, 36]. The indexing phase encompasses data cleansing, extraction, transformation into plain text, segmentation, and indexing, utilizing embedding models to transform text fragments into vector representations [2, 18]. In the retrieval phase, the system computes similarity scores based on the user's query to select the most pertinent text fragments. In the generation phase, the query and selected documents are amalgamated into prompts, facilitating the LLMs in

generating a response. While this method is straightforward, it encounters challenges related to retrieval quality, generation quality, and enhancement processes [21, 23].

In response to these challenges, researchers concentrate on the enhancement of the retriever, a task that can be categorized into three key aspects: pre-retrieval processing, retrieval model optimization, and post-retrieval processing [20]. Pre-retrieval processing encompasses data transformer to enhance text standardization, ensuring factual accuracy, optimizing index structures, adjusting block sizes, and rewriting query [4, 16, 50, 53]. Retrieval model optimization entails the fine-tuning of domain-specific embedding models and the application of dynamic embedding techniques [11, 63]. Post-retrieval processing minimizes context length through reranking and compression operations, aiming to emphasize critical information, diminish noise interference, and enhance integration and utilization by the generator [37, 55, 57].

Furthermore, to enhance the precision and efficiency of the generator when handling retrieval content, scholars have undertaken a series of optimization measures. As an illustration, researchers have devised methods such as **chain-of-note (CON)** for the generator [60]. CON generates continuous reading notes to comprehensively evaluate the relevance of retrieved documents to the posed question, integrating this information to produce precise final responses. This approach further enhances the capability of RAG in managing retrieval information, guaranteeing the production of responses that are simultaneously accurate and pertinent. In specific domains, such as medical and legal, models undergo fine-tuning to enhance the generator's performance within those particular fields [10, 24, 58]. Through the implementation of these methods, the generator can more effectively process retrieved information and furnish responses that are more accurate and relevant.

### 2.2 RAG Benchmarks

When investigating the development and optimization of RAG, the effective evaluation of their performance becomes a fundamental concern. Table 1 shows some commonly used benchmarks for evaluating RAG. LangChain provides benchmark tasks, such as LangChain Docs Q&A and Semi-Structured Reports [27], designed to assess various RAG architectures. These datasets are constructed from snapshots of Python documentation and PDFs containing tables and charts. They emphasize the model's capability to handle structured and semi-structured data. Evaluation standards encompass the accuracy of answers and the faithfulness of model responses. Utilizing large models for QA generation has emerged as a prevalent approach in building evaluation datasets. For instance, RGB [8] creates its evaluation dataset by gathering recent news reports and employing LLM to generate relevant events, questions, and answers. Conversely, ARES [44] relies on generating synthetic queries and answers, leveraging the FLAN-T5 XXL model. These methods not only showcase the RAG system's proficiency in handling real-time data but also illustrate the utility of automation and synthetic data in the evaluation process. For evaluating the capabilities of models across various professional domains, the Instruct-Benchmark-Tester dataset encompasses a range of question types, with a particular focus on financial services, legal, and intricate business scenarios [39].

Depending on whether the evaluation phase incorporates ground truth, metrics of existing evaluation methods can be categorized into those necessitating reference and those not requiring it. Reference-required evaluation methods gauge the accuracy and robustness of the RAG by contrasting model-generated answers with factual benchmarks. As an example, RAG-Instruct-Benchmark-Tester [39] employs accuracy score as an evaluation metric, a widely acknowledged measure of model performance that assesses the extent to which model-generated answers align with reference answers. The primary objective of RGB [8] is to evaluate whether large models can

Table 1. Related Work

| Method | Dataset | Scale | Evaluation Metrics | Evaluation Method | Application Field | Ref. | Lang. |
|---|---|---|---|---|---|---|---|
| [27] | Based on LangChain Python documentation QA dataset | 86 | Accuracy of answer, faithfulness of response to the retrieved document | Evaluating retrieval and generation consistency | General QA scenarios (*Read*) | Yes | EN |
| [27] | PDF documents containing tables and charts | 5 | Accuracy of answer, faithfulness of response to the retrieved document | Evaluating retrieval and generation consistency | Semi-structured data scenarios (*Read*) | Yes | EN |
| [32] | Query and responses (with citations) | 1,450 | Fluency, perceived utility, citation recall and precision | Human evaluation | Citation (*Read*) | Yes | EN |
| [17, 22] | Questions, answers and contexts (with citations) | - | Fluency, correctness, citation quality | Self-devised metrics, human evaluation | Citation (*Read*) | Yes | EN |
| [56] | Questions, answers and contexts (with citations) | 1,948 | Location citation recall, location precision, the coefficient of variation of citation locations | Self-devised metrics | Citation (*Read*) | Yes | EN |
| [29] | Questions, answers and contexts (with citations) | 3,422 | Accuracy of factual and non-factual, AUC-PR, and so on | Self-devised metrics, common metrics | Citation (*Read*) | Yes | EN |
| [64] | Statement-citation pairs | 12,681 | Correlation, classification performance, retrieval effectiveness, faithfulness | Self-devised metrics, common metrics, human evaluation | Citation (*Read*) | Yes | EN |
| [7] | Paragraphs (with citations) | 10,000 | Citation density, the coverage of reference facts | Self-devised metrics | Citation (*Read*) | Yes | EN |
| [39] | Questions, answers and contexts | 200 | Categorization ability, logical/mathematical reasoning, complex question solving, summarization ability | Accuracy | Financial services, legal, business (*Read*, *Delete*) | Yes | EN |
| [8] | LLM-generated dataset | 1,000 | Noise robustness, negative rejection, information integration, counterfactual robustness | Self-devised metrics | General, especially news domain (*Read*, *Update*) | Yes | CN EN |
| [14] | – | – | Context relevance, groundedness, answer relevance | Analyzing the RAG triad | General (*Create*, *Read*) | No | – |
| [13] | – | – | Faithfulness, answer relevance, context relevance | Automated evaluation using LLM prompts | General (*Create*, *Read*) | No | – |
| [44] | LLM-generated dataset | 150 | Context relevance, answer faithfulness, answer relevance | Generating custom LLM judges for each component of a RAG system | General (*Create*, *Read*) | No | EN |
| *Ours* | LLM-generated dataset | 36,166 | ROUGE, BLEU, bertScore, RAGQuestEval | Evaluating retrieval and generation consistency | General (*Create*, *Read* *Update*, *Delete*) | Yes | CN |

AUC-PR, area under the precision-recall curve. Bold numbers indicate the best results.

effectively utilize external documents to acquire knowledge and generate accurate answers. Its evaluation metrics encompass accuracy, rejection rate, error detection rate, and correction rate.

Reference-free evaluation methods, including TruLens-Eval [14], RAGAS [13], and ARES [44], provide distinct viewpoints for evaluating the performance of RAG systems, particularly concerning context relevance, answer faithfulness, and answer relevance. TruLens-Eval [14] introduces the RAG Triad as an innovative approach to evaluate hallucination issues within the RAG architecture, encompassing context relevance, groundedness, and answer relevance. RAGAS [13], serving as a reference-free evaluation framework, concentrates on assessing the retrieval system's capacity to identify pertinent and concentrated context passages, along with the LLMs' proficiency in faithfully and accurately leveraging these passages. In contrast to RAGAS, which depends on a predefined set of heuristically crafted prompts, ARES generates tailored LLMs judges for each aspect of a RAG pipeline, leading to a substantial enhancement in evaluation precision and accuracy when compared to existing methods such as RAGAS. Furthermore, ARES [44] employs prediction-powered inference to offer statistical assurances for its scoring, generating confidence intervals. ARES emphasizes three evaluation scores: context relevance, answer faithfulness, and answer relevance, highlighting the importance of a proficient RAG system in identifying relevant contexts and producing both faithful and relevant answers. Regarding evaluation methods, Liu et al. [32] place an emphasis on assessing the credibility and accuracy of responses generated by generative search engines through manual inspection. Nonetheless, manual evaluation possesses drawbacks, including high costs and

challenges in scalability. Hence, rule-based evaluation metrics such as accuracy, exact match, rouge, or self-devised metrics like rejection rate, error detection rate, and correction rate continue to be widely adopted in the field. Furthermore, employing LLMs for evaluation closely approximates manual evaluation outcomes.

### 2.3 Citation-Enhanced RAG

In traditional RAG methods, despite the rich information sources provided by retrieved contexts for text generation, these models often do not explicitly require responses to provide corresponding citations, making traceability difficult. Therefore, enhancing text verifiability by introducing citation links, i.e., explicit references, has become an important research direction in the RAG field [17, 56].

Providing citation indicators in the response text offers several clear benefits. First, users can easily verify the claims made by LLMs based on the provided citations, thus improving the transparency and credibility of the text. Second, if the text generated by LLMs adheres faithfully to the cited contexts, it can significantly improve its accuracy and reduce the phenomenon of "hallucinations" [17]. Given this, generating high-quality citations and evaluating the quality of citation generation have become crucial elements of assessing RAG performance. Constructing appropriate prompts directly through the retrieval context to guide the model in generating corresponding citations constitutes a direct and effective method of citation generation [22].

In terms of evaluation, early research primarily focused on the fluency, accuracy, and basic citation quality of the text generated by LLMs [17, 29]. For example, Rashkin et al. proposed the "Attributable to Identified Sources" score [42], which serves as a valuable tool for measuring the degree to which generated text is faithful to its sources. As research progressed, scholars recognized the need for more detailed evaluation methods to differentiate between various levels of citation support. By creating specialized datasets such as SCIFI [7], researchers can more precisely evaluate fine-grained citations at the clause level in texts generated by LLMs. The ALiiCE framework [56], by analyzing the atomic structure of sentence claims, introduced fine-grained evaluation metrics, such as location citation recall and precision, and the coefficient of variation of citation locations, to more granularly evaluate the quality of citation generation in RAG [56]. In practical applications, Zhang et al. [64] found that RAG requires more complex evaluation frameworks to distinguish between various levels of citation support by comparing different fidelity metrics. These RAG evaluation methods not only consider the presence of citations but also their accuracy and relevance.

While Citation-Enhanced RAG delves deeply into the specific domain of citation generation, aiming to improve the credibility and accuracy of text generated by RAG systems, our benchmark provides a comprehensive evaluation framework encompassing various aspects of RAG systems and multiple application scenarios.

## 3 CRUD-RAG: A Comprehensive Chinese Benchmark for RAG

As we discussed earlier, implementing RAG effectively requires careful tuning of multiple components, such as the retrieval model, the knowledge corpus, the language model, and the query formulation. Therefore, we need a framework that can evaluate the RAG system automatically. This framework would enable us to examine how these components affect the system's performance and provide us with useful insights for improving and innovating the system.

However, the current RAG benchmarks have several drawbacks: they only evaluate question-answering tasks [1, 41, 59], ignoring other diverse application of RAG. The optimization strategy for QA tasks may not suit other tasks; in the evaluation experiment, current RAG benchmarks only account for the LLM component in the RAG pipeline, or the retriever in the knowledge-intensive

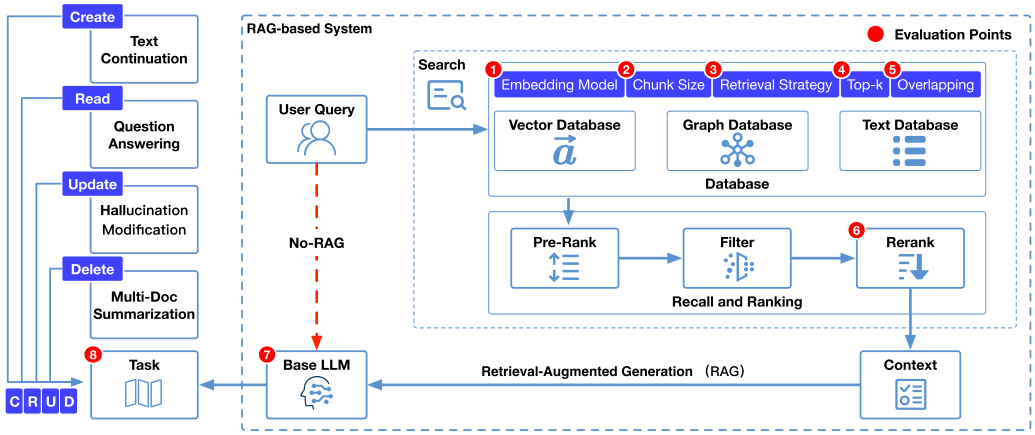


Fig. 2. Illustration of CRUD-RAG, our comprehensive Chinese benchmark for RAG. It classifies the RAG application scenarios into four categories: create, read, update, and delete. For each category, we create appropriate evaluation tasks and datasets. In the experiments, we evaluate various components of the RAG system using our benchmarks.

scenario, disregarding the vital roles of retrieval database construction and retrieval in non-knowledge-intensive scenarios.

To address the shortcomings of previous benchmarks, we introduce CRUD-RAG, a comprehensive Chinese benchmark for RAG. Figure 2 illustrates the features of our CRUD-RAG benchmark. It classifies the RAG application scenarios into four categories: create, read, update, and delete, then we construct appropriate evaluation tasks and datasets for each category. Besides, in the experiments, we will assess the impact of various components of RAG, such as chunk size, retrieval strategy, top-k, LLM, and so on, on all tasks.

In the following section, we will describe the evaluation tasks and the datasets that we design for each RAG application scenario type. We select text continuation, QA (single and multi-document), hallucination modification, and multi-document summarization as representative tasks in the CRUD scenario and construct corresponding datasets. The summarization (D) and continuation (C) datasets were constructed simultaneously, since the construction of both datasets requires the use of a search engine. They will be discussed together in the following section. The construction of the QA (R) and hallucination modification (D) datasets is relatively independent. To maintain narrative coherence, we will introduce the dataset construction process in the order of DCRU. Table 2 presents the size and composition of our datasets, and Figure 3 illustrates an example of our datasets.

### 3.1 News Collection

As mentioned above, the existing benchmarks for evaluating RAG systems are mainly constructed for QA tasks. Therefore, the datasets, such as NQ [26] and RGB [8], are also tailored for this type of task. Hence, we need to construct new datasets.

We argue that the latest news data is the most suitable choice for creating an RAG evaluation dataset. Unlike other types of data, such as encyclopedias, questions, or conversations, the latest news minimizes the possibility that the model has been exposed to similar content during training. This dependency on external retrieval mechanisms allows for a comprehensive evaluation of the entire RAG process, not just the model's generation ability. Additionally, news data is easy to collect, enabling us to maintain dataset timeliness. When the existing dataset loses its timeliness,

Table 2. The Composition of Our Datasets

| Dataset Name | Dataset Size | Components | Evaluation Objectives |
|---|---|---|---|
| Text continuation | 10,728 | An initial part of an article, followed by its extension or completion. | Evaluate the RAG system's performance in "Create" scenarios (creative generation). |
| QA (1-document) | 3,199 | A collection of question-answer pairs, where the answer is directly extractable from a document passage. | Evaluate the RAG system's performance in "Read" scenarios (knowledge-intensive application). |
| QA (2-document) | 3,192 | A collection of question-answer pairs, where the answer *requires synthesis of information from two different document sources.* | The objective is the same as 1-document QA, but it also examines *the reasoning ability of combining two documents.* |
| QA (3-document) | 3,189 | A collection of question-answer pairs, where the answer *requires synthesis of information from three different document sources.* | The objective is the same as 1-document QA, but it also examines *the reasoning ability of combining three documents.* |
| Hallucination modification | 5,130 | Some sentences containing errors, and the sentence with the errors fixed. | Evaluate the RAG system's performance in "Update" scenarios (error correction application). |
| Multi-doc summarization | 10,728 | A one-sentence headline of an article, followed by a brief summary of the article. | Evaluate the RAG system's performance in "Delete" scenarios (summarization). |
| Retrieval database | 86,834 | As the knowledge base for the RAG system, we expect the RAG system to retrieve relevant content from the knowledge base to address the above tasks. | ——— |

Bold numbers indicate the best results.

we can quickly gather the latest news to rebuild a more challenging dataset. Moreover, the latest news data offer rich and diverse topics and content, which can test the model's performance and adaptability in various domains and situations.

Therefore, we select news as the base of our datasets. To ensure the authenticity and currency of the datasets, we collected nearly 300,000 of historical news articles from major Chinese news websites published after July 2023, which were not exposed to the LLMs during the training phase. We remove duplicate news documents from 300,000 news articles and filter out those that are too long or too short. We ended up with more than 80,000 news articles. Based on the news corpus we collected, we constructed our datasets for three tasks, namely open-domain multi-document summarization, text continuation, and QA.

### 3.2 Open-Domain Multi-Document Summarization: RAG Application in "Delete"

In one of the RAG's application scenarios, "Delete," the RAG system retrieves key information from external sources based on the input text, and eliminates redundancy and irrelevance, to generate concise summaries. A suitable task for evaluating this scenario is multi-document summarization, which aims to generate a brief and coherent summary from a set of related documents. For the news data we collect, this task involves retrieving major media reports on a news event, and summarizing the background, process, and results of the event.

However, constructing such a dataset is extremely challenging. First, news articles retrieved based on events may not be fully relevant, requiring manual filtration to identify the correct and

| Dataset | Query | Best Matching Context | Ground Truth Reference |
|---|---|---|---|
| Multi-Doc Summarization | 2023年7月31日，由于匈牙利执政党的缺席，匈牙利国会未能就批准瑞典加入北约的提案进行投票，导致瑞典加入北约的申请再次被推迟。<br><br>On July 31, 2023, due to the absence of Hungary's ruling party, the Hungarian Parliament failed to vote on the proposal to approve Sweden's accession to NATO, resulting in another delay of Sweden's NATO membership application. | —— | 2023年7月31日，匈牙利国会在特别会议上未能就批准瑞典加入北约的提案进行投票，原因是拥有三分之二多数席位的执政党成员未能出席，由此瑞典加入北约的申请再次被推迟…<br><br>On July 31, 2023, the Hungarian Parliament failed to vote on the proposal to approve Sweden's accession to NATO during a special session, due to the absence of members from the ruling party, which holds a two-thirds majority. As a result, Sweden's NATO membership application was delayed once again... |
| Text Continuation | 昨晚，工信部发布组织开展移动互联网应用程序备案工作的通知，从9月起组织开展APP备案工作…<br><br>Last night, the Ministry of Industry and Information Technology issued a notice to organize the registration of mobile internet applications, starting the APP registration process from September... | —— | 工信部表示，综合考虑APP主办者、网络接入服务提供者、应用分发平台、智能终端生产企业实际业务情况，预留10个月时间作为APP备案工作的过渡期…<br><br>The Ministry of Industry and Information Technology stated that, taking into account the actual business situations of APP operators, network access service providers, application distribution platforms, and smart terminal manufacturers, a 10-month transition period will be reserved for the APP registration process... |
| Question Answering (1-document) | 陕西西安市最近发放的体育类电子消费券的具体金额是多少？可以在多少家体育场馆使用？<br><br>Recently, Xi'an, Shaanxi Province, has issued a large number of sports electronic coupons for citizens. What is the specific amount of these electronic coupons? In how many sports stadiums can they be used? | 近日，陕西西安市发放500万元体育类电子消费券，市民领取后，可在全市173家体育场馆使用。体育消费券的发放，激发了群众参与健身的热情，同时也带动了体育场馆的运营 …<br><br>Recently, Xi'an City in Shaanxi Province distributed 5 million yuan worth of sports electronic vouchers. After receiving these vouchers, citizens can use them at 173 sports venues across the city. The distribution of sports vouchers has stimulated the public's enthusiasm for fitness and also boosted the operation of sports venues... | 西安市发放500万元体育消费券，市民可在173家场馆使用<br><br>Xi'an City has distributed 5 million yuan worth of sports vouchers, which citizens can use at 173 venues. |
| Question Answering (2-document) | 考虑到美国大豆生长的最新数据、短期极端天气预报以及豆粕需求和库存情况，美豆期货价格在短期内将如何变化？<br><br>Considering the latest data on U.S. soybean growth, short-term extreme weather forecasts, and the current demand and inventory of soybean meal, how will U.S. soybean futures prices change in the short term? | 1. 本周美国农业部公布的数据显示，截至2023年7月23日当周，美国大豆…尽管优良率数据不及预期，但其他关键数据则表现较好，其中结荚率已经开始加快增长… This week, data released by the U.S. Department of Agriculture showed that as of the week ending July 23, 2023, U.S. soybeans ... This week's data has surpassed both the same period last year and the five-year average, indicating a faster growth rate of soybeans...<br>2. 美国国家气象局称，这一轮极端热浪将至少持续到7月28日，预计接下来几天热浪范围还将扩大到美国中西部的许多地区，届时当地气温可能会达到今年的最高水平。短期的热浪天气，对结荚期的大豆影响相对有限…The U.S. National Weather Service stated that ... short-term heatwaves will have a relatively limited impact on soybeans during the pod-setting stage .. | 尽管美国大豆的优良率略低于市场预期，但结荚率的增长超过了去年同期和五年均值，表明大豆生长速度较快，市场无需担忧减产，产量预估稳定。同时，短期内极端热浪天气对结荚期的大豆影响有限，而多瑙河附近的国际农产品出口形势的不确定性为美豆期货提供了上涨支撑。另一方面…<br><br>Although the quality of U.S. soybeans is slightly below market expectations, the pod-setting rate has surpassed both the same period last year and the five-year average, indicating a faster growth rate. The market need not worry about a production decrease, and the yield estimates remain stable. At the same time, short-term extreme heatwave weather has limited impact on soybeans during the pod-setting stage. The uncertainty in the international agricultural product export situation near the Danube River provides upward support for U.S. soybean futures. On the other hand... |
| Hallucination Modification | 习近平会见沙特王储继承人穆罕默德，习近平指出，中沙建交25年来，两国关系始终保持健康稳定发展势头，各领域合作不断深化。<br><br>Xi Jinping met with Saudi Crown Prince Mohammed. Xi Jinping pointed out that in the 25 years since the establishment of diplomatic relations between China and Saudi Arabia, the relations between the two countries have always maintained a healthy and stable development momentum, with cooperation in various fields continuously deepening. | 习近平指出，中沙建交26年来，双方始终真诚友好、平等相待，各领域合作取得许多成果。中国视沙特为共建“一带一路”的重要合作伙伴，愿同沙方共同努力…<br><br>Xi Jinping pointed out that in the 26 years since the establishment of diplomatic relations between China and Saudi Arabia... China regards Saudi Arabia as an important partner in building the "Belt and Road" and is willing to work with Saudi Arabia to enrich ... | 习近平指出，中沙建交26年来，两国关系始终保持健康稳定发展势头，各领域合作不断深化<br><br>Xi Jinping met with Saudi Crown Prince Mohammed. Xi Jinping pointed out that in the 26 years since the establishment of diplomatic relations between China and Saudi Arabia, the relations between the two countries have always maintained a healthy and stable development momentum, with cooperation in various fields continuously deepening. |

Fig. 3. Some examples of the datasets we constructed. We did not provide the best matching context for the multi-document summarization dataset and the text continuation dataset, because these two datasets were built in a reverse way, and the context matching degree for these two tasks was rather vague.

pertinent documents. Then, when generating summaries from these documents, it is essential to eliminate a significant amount of redundant information, retaining only the most important content. These tasks require manual annotation, which consumes substantial time and financial resources, and often results in too much redundant information.

Fortunately, we can use an existing method, which constructs a multi-document summary dataset in reverse [34]. Figure 4 shows the construction process of multi-document summarization. In particular, our dataset construction process is as follows:

—Instead of generating event summaries based on multiple related news content, we first acquire a news article from a high-quality corpus, and annotate its summary and events.
—Then, we search for external reference materials related to the current news by using the event text, ensuring they are connected but not the same. We conduct extensive searches to gather sufficient information to reconstruct the summary of the selected news.
—In this manner, the reference literature we collect, along with the summary of the current news, collectively form a dataset of multi-document summarization.

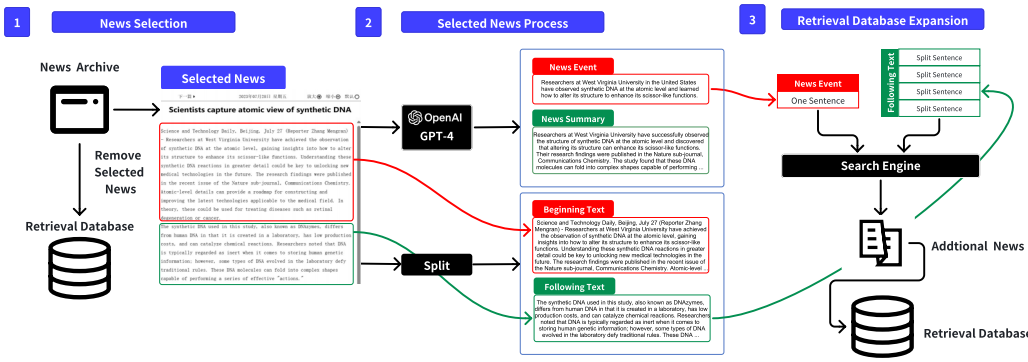


Fig. 4. The dataset construction pipeline for text continuation and multi-document summarization task.

Specifically, we first select 10,000 news articles $d$ from our high-quality news corpus $D$, and then use GPT-4 to generate summaries and events for each article. Next, we use the events as keywords, and search for the most relevant 10 news articles on Baidu, excluding any data that is too similar to the original article. We repeat this process for all the articles, and add the expanded articles to our news corpus, removing the 10,000 articles $d$ simultaneously. The new news corpus $D - d + E$ serves as our retrieval corpus, and we expect the model to use the events and relevant information from the retrieval corpus to generate a summary of the articles $d$.

### 3.3 Text Continuation: RAG Application in "Create"

RAG is useful not only for "Delete," where it retrieves and summarizes key information from massive texts, but also for "Create." In this scenario, RAG systems show strong creativity by expanding existing texts, and we take the text continuation task as an evaluation. The text continuation task aims to automatically produce coherent and relevant subsequent content based on the beginning of the text, making the text more complete and vivid.

To construct the continuation task dataset, we follow the same method as the summary task dataset. Figure 4 shows the construction process of text continuation. Specifically, we select a news article from a high-quality corpus and use a specialized Chinese word segmentation tool, to split it into sentences. Then, we divide the article into two equal parts: the first half serves as the input, and the second half as the output of the continuation dataset. We expect the model to use RAG technology to retrieve relevant information from the document library and generate a continuation that is coherent, informative, and consistent with the input and output.

To ensure that the retrieval database covers the real continuation text, we use the Baidu search engine to find external documents and add them to the database. The continuation text differs from the event text in that it consists of multiple sentences. Therefore, we split the continuation text into paragraphs by sentences and retrieve relevant documents for each paragraph using the search engine. This way, we guarantee that the retrieval database contains most of the information to reconstruct the continuation text.

### 3.4 QA: RAG Application in "Read"

Another application scenario of RAG is to use external knowledge bases to enhance the question-answering capabilities of LLMs, which can be applied to various knowledge-intensive tasks. Currently, there are many evaluation benchmarks to measure the performance of RAG in this scenario, and multiple QA datasets have been created.


Fig. 5. The dataset construction pipeline for multi-document (inferential) QA task.

However, the existing QA datasets also have some limitations. On the one hand, some datasets (such as NQ and WEBQA) are outdated, and may have been covered by LLMs in the pre-training stage, which reduces the advantage of RAG systems. On the other hand, some datasets (such as RGB) only contain some factual questions, which can be directly extracted from the retrieved texts, without requiring complex reasoning over multiple texts, which poses less challenge to RAG systems. The most recent LLMs capture enough knowledge to rival human performance across a wide variety of QA benchmarks [5].

To overcome these limitations, we build a large-scale QA dataset, which is divided into two parts: single-document and multi-document QA. Single-document QA focuses on factual questions that ask for specific details in the news, such as the location or the main characters of an event. Multi-document QA, on the other hand, involves inferential and critical thinking questions that require readers to reason across multiple news paragraphs, such as comparing and contrasting two events or assessing their impact.

For the single-document QA task, we follow the dataset construction process of the previous RGB benchmark [8]. We first select news articles from our collected high-quality corpus. Then we use prompts to make GPT-4 generate questions and answers for each article. For example, for a report on "The 2023 Nobel Prize," GPT-4 will generate the question "Who was awarded the 2023 Nobel Prize for Physiology and Medicine?" and provide key information for answering it.

For the multi-document QA task, constructing a reasoning question that requires the synthesis of multiple documents is not trivial. Simply using a prompt to force GPT-4 to generate the question is ineffective, because creating such a multi-document QA dataset is a complex reasoning task in itself. Therefore, we adopt CoT technology [51] to enhance GPT-4. We guide the model to build the dataset gradually through multiple reasoning steps. Figure 5 illustrates our specific process for building a two-document QA dataset using GPT-4 and CoT technology. We will explain it in detail:

(1) *Retrieve multiple connected news*, which should cover the same event, but offer different perspectives or information.
(2) *Use prompts to help GPT-4 identify the common elements between different reports*, such as the event they report on, and ensure they are relevant.

(3) *Use prompts to help GPT-4 distinguish the differences between news articles*. While keeping the connection between reports, we analyze the differences between each report. This step requires comprehensive understanding and analysis from multiple angles and avoids generating questions that can be answered from a single paragraph.
(4) *Generate the question based on different focus points*, which should require integrating information from multiple sources to answer.
(5) *Reconstruct the question based on the contact point*. Based on the connections in the reports, refine the questions, ensuring the inherent logical connection, and avoiding superficial combinations. The questions should be logically linked, rather than physically juxtaposed. For example, instead of simply asking "Describe the history of World War II and explain the basic principles of quantum physics," a question like "How did the technological and political environment during World War II foster the development of quantum physics?" should be formulated, where the parts are interdependent or have causal relationships.

We constructed two types of multi-document QA datasets with different levels of difficulty: one requires reasoning from two documents to answer the question, and the other is more challenging and requires reasoning from three documents to answer the question.

To further ensure our dataset's quality, we employed a manual refinement process for the data generated by GPT-4. Our annotation team comprises three native Chinese speakers, each with at least a bachelor's degree. The annotation process is as follows:

(1) The annotator evaluates the quality of the automatically generated query and chooses one of the following two options:
—*Reasonable*: Conforms to natural language usage.
—*Needs refinement*: Has issues with naturalness, accuracy, or grammar.
(2) If "Reasonable" is selected, no further action is taken. If "Needs refinement" is chosen, the annotator manually improves the query's naturalness and accuracy.

In addition to their standard salary, annotators receive an extra 1 RMB per query evaluated or refined. The average annotation time per query is approximately 20 seconds. To ensure annotation quality, we randomly inspected 5% of the annotated data.

Given the substantial cost of manual annotation and the large size of our dataset, we initially polished one-fifth of our dataset manually. We will continuously monitor dataset quality across various social media platforms and refine it manually as needed.

Notably, only 5.8% of queries required refinement, indicating that the queries generated by GPT-4 are generally of high quality. This validates the effectiveness of using GPT-4 for initial data generation and underscores our commitment to ensuring dataset quality.

### 3.5 Hallucination Modification: RAG Application in "Update"

Besides the three scenarios mentioned above, the RAG framework can also be used to correct errors in the text. This involves using the RAG framework to access relevant information from external sources, identify and correct errors in the text, and maintain the accuracy of the text content.

We construct a hallucination modification dataset using the open source large-scale dataset UHGEval [31]. UHGEval instructs the model to generate continuations that contain hallucinations for a given news text. It utilizes GPT-4 for automatic annotation and human evaluation to identify and mark segments in the text containing hallucinations. In our approach, we input the hallucination text along with the corresponding annotations from the dataset. Subsequently, GPT-4 is employed to rectify the hallucinations, resulting in the production of the text without any hallucinatory elements. Finally, real news continuations will be included in the document retrieval database.

The RAG system's experimental results on this dataset can confirm if the system can retrieve the real news information from the document database based on the input text, which consists of the beginning text and the hallucination continuation text, and then correct the hallucination text to generate the text without hallucination.

### 3.6 Evaluation Method

The aim of this benchmark is to evaluate how well RAG systems can retrieve relevant documents, and use them to generate sensible responses. Therefore, we adopt an end-to-end evaluation method, which directly compares the similarity between the model output and the reference answers.

Evaluating the performance of RAG systems requires choosing appropriate evaluation metrics. We considered the previous evaluation metrics for text generation, *ROUGE* and *BLEU*, which are both based on word overlap. ROUGE mainly counts the recall rate on the n-gram, while BLEU mainly counts the precision rate on the n-gram. However, BLEU and ROUGE are word overlap-based accuracy metrics that depend on the overall expression of the text, and do not capture the accuracy of the particular key information in the text. Therefore, they may not reflect the factual consistency of a text well, especially for long texts. To alleviate this issue, recent work [12, 45, 49] has proposed new evaluation metrics for abstractive summarization evaluation. These metrics are based on the intuition that if you ask questions about the summary and the original document, you will get a similar answer if the summary realistically matches the original document. They evaluate the accuracy of each local piece of key information in the summary.

We also consider QA-based metrics to evaluate the factual accuracy of generation. In this article, we examine *QuestEval* [45], a metric that improves the correlation with human judgments over previous metrics in their extensive experiments. QuestEval evaluates the factual consistency between the generated text and the source document, which is mainly used for text summarization tasks. Therefore, it does not require any ground truth reference. However, for RAG systems, the retrieved texts may be irrelevant or incorrect, so consistency with them is not a valid criterion. Instead, we use this metric to measure how well the generated text matches the ground-truth reference. We call this metric *RAGQuestEval*. We will explain this metric in detail.

As Figure 6 displayed, let $GT$ and $GM$ be two sequences of tokens, where $GT$ denotes the ground truth references and $GM$ the corresponding evaluated generations. First, we generate a series of questions from the ground truth references $GT$ using the QuestEval method, which extracts entities and noun phrases from the text. The goal of *RAGQuestEval* is to check if the generated text includes and conveys correctly all the key information from the ground truth reference.

Next, we answer these questions using both real references and model-generated text. If the question is unanswerable, the model returns "<Unanswerable>."

Finally, we calculate two scores to evaluate the quality of the generated text: recall and precision.

*Recall.* Recall is the ratio of answerable questions to all questions. This score shows how much information in the ground truth reference is captured by the text generated by the RAG system. A higher recall means that the generated text covers more information from the reference

$$\text{Recall}(GT, GM) = \frac{1}{|Q_G(GT)|} \sum_{(q,r)\in Q_G(GT)} \mathbb{I}[Q_A(GM, q) \neq \text{<Unanswerable>}]. \tag{1}$$

In the above equation, $Q_G$ is the question generator and $Q_A$ is the question answerer.

*Precision.* Precision is the average answer similarity of all questions, excluding the unanswerable ones. We use the token level F1 score to measure the answer similarity, which is a standard metric


Fig. 6. Overview of RAGQuestEval. A set of questions is generated based on the ground truth references. The questions are then answered using both the ground truth and the response. For the recall score of RAGQuestEval, we calculate the ratio of answerable questions to all questions (in this case, recall = 2/3). For the precision score of RAGQuestEval, corresponding answers are compared using a similarity function and averaged across questions (in this case, precision = (0.5 + 1)/2 = 0.75). The recall metric of RAGQuestEval indicates how much of the key information in the ground truth reference is included in the generated text, while the precision metric of RAGQuestEval indicates how correct the recalled key information is.

for evaluating factoid QA models. Higher precision means that the generated text is more accurate and consistent with the reference

$$\text{Prec}(GT, GM) = \frac{1}{|Q_G(GT)|} \sum_{(q,r)\in Q_G(GT)} F1\left(Q_A(GM, q), r\right). \tag{2}$$

## 4 Experiment

The current evaluation of RAG benchmark only focuses on the LLM component in the RAG pipeline and overlooks the importance of retrieval database construction and retriever. To address this gap, we examine how different aspects of RAG systems affect their performance in our benchmark. We also discuss some possible ways to improve existing RAG systems.

### 4.1 Experimental Settings

In this section, we will introduce the components of the RAG system and describe how we conduct experiments to evaluate their impact on system performance. The RAG system consists of the following components:

— *Chunk size*: The RAG system splits the external knowledge into chunks of a certain length and stores them in a vector database. The chunk size affects the retrieval accuracy and the completeness of the context.
— *Chunk overlap*: Chunk overlap refers to the shared tokens between two consecutive text chunks and is used to ensure semantic coherence when chunking.
— *Embedding model*: The RAG system converts the text chunks and the user's query into vectors using an embedding model or other methods. The embedding model affects the quality and relevance of the context.

—*Retriever*: The RAG system uses a retriever to find the top-k vectors most similar to the query vector in the vector database and retrieves the corresponding text chunks. The retriever affects the richness and diversity of the context.
—*Top-k*: This is the number of text chunks that the RAG system retrieves for each query, which serves as the context part of the LLMs prompts. The top-k influences the size of the context that the model receives.
—*LLM*: The RAG system inputs the context and the query to an LLM to generate the answer. The LLM affects the correctness and rationality of the answer.

We use the following settings as the basic version of our RAG system: chunk size: 128, chunk overlap: 0%, embedding model: bge-base, retriever: dense retriever, top-k: 8, and LLM: GPT-3.5. In the experiments, we change one component at a time and evaluate the results on different tasks. We compare the following values for each component:

—Chunk size: 64, 128, 256, 512.
—Chunk overlap: 0%, 10%, 30%, 50%, 70%.
—Embedding model: m3e-base, bge-base, stella-base, gte-base.
—Retriever: dense, bm25, hybrid, hybrid+rerank.
—Top-k: 2, 4, 6, 8, 10.
—Base LLMs: GPT-3.5, GPT-4, ChatGLM2-6B, Baichuan2-13B, Qwen-7B, Qwen-14B.

In the experiments, we use two types of evaluation metrics: The overall semantic similarity metrics (bleu, rouge-L, and bertScore) measure how closely the generated content matches the reference content in terms of meaning and fluency; and the key information metric (RAGQuestEval) measure how well the generated content captures and presents the key information from the reference content.

Considering that we used gpt-3.5 as the baseline model for the experiments, to reduce the cost, we only conducted experiments on 1/5 of our dataset.

### 4.2 Analyzing the Impact of Chunk Size on RAG Performance in Different Tasks

Chunking is the process of dividing a document into chunks of a fixed length, and then converting each chunk into a vector and storing it in an index. This creates an external knowledge index. Chunk size is a crucial parameter that varies depending on the corpus characteristics. If the chunk is too small or too large, it can reduce the search accuracy or omit important content. Hence, finding the optimal chunk size is vital for ensuring search accuracy and relevance, and enabling the LLMs to generate appropriate responses. Our experiments reveal that different RAG tasks correspond to different optimal chunk sizes.

*Text Continuation.* The experimental results in Table 3 demonstrate that larger chunk size can improve the overall semantic similarity measures (bleu, rouge-L). Besides, the RAGQuestEval metrics, which reflect the precision and recall rate of key information, follow a consistent pattern. This indicates that larger blocks preserve the original document's structure, which is crucial for creative tasks such as text continuation. Smaller chunks, on the other hand, result in fragmented and semantically incoherent content, which impairs the ability of large models to understand and generate engaging content.

*Open-Domain Multi-Document Summarization.* We observe some intriguing patterns in the experimental results. Firstly, we discover that larger chunk size not only substantially increases the length of the generated text but also cause a notable drop in the bleu score, while the rouge-L and bertScore remain almost unchanged. This implies that larger chunks can preserve more original text information, but also introduce some semantic redundancy. Secondly, for the RAGQuestEval

Table 3. The Experimental Results for Evaluating Different Chunk Sizes in Our Benchmark

| Task Name | Chunk Size | Top-k | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Precision | Recall | |
| Text continuation | 64 | 16 | 3.42 | 17.67 | 83.94 | 26.09 | 23.39 | 345.8 |
| | 128 | 8 | 3.66 | 17.78 | 83.99 | 26.96 | 24.68 | 367.6 |
| | 256 | 4 | 4.21 | 17.93 | **84.17** | 28.86 | 25.99 | 403.0 |
| | 512 | 2 | **5.12** | **18.81** | 83.57 | **30.91** | **28.27** | 413.2 |
| Summarization | 64 | 16 | **24.60** | 33.78 | 88.07 | **68.29** | 43.98 | 184.2 |
| | 128 | 8 | 23.69 | 33.53 | 88.49 | 68.06 | 46.18 | 205.9 |
| | 256 | 4 | 22.97 | **33.85** | 88.83 | 67.87 | 48.66 | 219.9 |
| | 512 | 2 | 21.08 | 33.23 | **88.89** | 66.43 | **50.31** | 243.6 |
| QA 1-document | 64 | 16 | 37.50 | 55.45 | 83.02 | 48.31 | 68.62 | 71.5 |
| | 128 | 8 | **39.76** | **57.24** | 83.81 | 52.67 | 70.82 | 73.3 |
| | 256 | 4 | 38.43 | 56.20 | **84.02** | **52.83** | **72.21** | 79.6 |
| | 512 | 2 | 36.51 | 54.64 | 82.72 | 51.26 | 68.65 | 84.1 |
| QA 2-document | 64 | 16 | 19.86 | 34.80 | 86.14 | 37.77 | 52.60 | 143.1 |
| | 128 | 8 | 22.75 | 37.25 | 87.16 | 42.93 | 56.73 | 149.8 |
| | 256 | 4 | **24.38** | 39.36 | 88.18 | 48.45 | 61.75 | 164.5 |
| | 512 | 2 | 24.05 | **39.69** | **88.22** | **49.24** | **63.37** | 176.7 |
| QA 3-document | 64 | 16 | 18.55 | 33.39 | 86.85 | 34.91 | 47.95 | 146.1 |
| | 128 | 8 | 21.05 | 35.04 | 87.81 | 40.32 | 51.37 | 156.6 |
| | 256 | 4 | **21.63** | 36.03 | 88.10 | 42.55 | 53.80 | 171.2 |
| | 512 | 2 | 21.40 | **36.55** | **88.38** | **44.28** | **57.38** | 183.6 |
| Hallucination modification | 64 | 16 | **34.20** | **54.90** | **81.14** | 64.98 | 80.96 | 60.7 |
| | 128 | 8 | 32.35 | 53.04 | 80.49 | **65.07** | 80.85 | 64.8 |
| | 256 | 4 | 31.48 | 51.76 | 80.15 | 64.93 | **80.99** | 67.7 |
| | 512 | 2 | 30.35 | 50.50 | 79.66 | 64.83 | 79.17 | 66.6 |

We use two types of evaluation metrics: the overall semantic similarity metrics (bleu, rouge-L, and bertScore) and the key information metric (RAGQuestEval). Bold numbers indicate the best results.

metric that evaluates key information, we found that a larger chunk size considerably enhances the recall of key information, but also lowers the precision of key information.

We hypothesize that this is because larger chunks enable the retrieval of more relevant content, thus improving the recall of key information. However, larger chunks also make the summarization task more challenging, as more fine-grained selection is required from the more relevant information, leading to lower precision of key information, which may not be a good thing for the summary.

*QA.* For single-document QA, too large chunks will reduce both recall and precision score of key information. The task only requires extracting information from a sub-paragraph of a single document, and the answer may be in a specific sentence. Therefore, smaller chunks are more suitable, as excessive content will make the extraction harder for the model.

For multi-document QA, the results are different from those of single-document QA. Larger chunks can significantly improve the recall and precision of key information, as well as the semantic similarity of the generated and reference answers. This is because larger chunks retain the original structure of the article, which is crucial for reasoning and understanding tasks, and fragmented information is not conducive to reasoning.

*Hallucination Modification.* For the hallucination modification task, the results are similar to those of the single-document QA task. Smaller chunks can significantly improve the semantic

Table 4. The Experimental Results for Evaluating Different Chunk Overlap Values in Our Benchmark

| Task Name | Chunk Overlap (%) | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|
| | | | | | Precision | Recall | |
| Text continuation | 0 | 3.66 | 17.78 | 83.99 | 26.96 | 24.68 | 367.6 |
| | 10 | 3.86 | 17.84 | 84.03 | 27.18 | 24.21 | 359.2 |
| | 30 | 3.91 | 17.92 | **84.12** | 28.21 | 24.72 | 367.0 |
| | 50 | 3.94 | 17.86 | 84.01 | **28.34** | 24.48 | 365.4 |
| | 70 | **4.03** | **17.95** | 84.04 | 27.64 | **25.32** | 364.0 |
| Summarization | 0 | 23.69 | 33.53 | 88.49 | 68.06 | 46.18 | 205.9 |
| | 10 | 23.54 | 33.59 | 88.35 | **68.67** | 46.16 | 208. 4 |
| | 30 | 23.74 | 33.58 | 88.41 | 68.02 | 46.08 | 203.3 |
| | 50 | 24.05 | 33.99 | 88.62 | 68.61 | 46.64 | 204.2 |
| | 70 | **24.49** | **34.29** | **88.71** | 68.45 | **47.08** | 201.8 |
| QA 1-document | 0 | **39.76** | 57.24 | 83.81 | 52.67 | 70.82 | 73. 3 |
| | 10 | 39.36 | **57.59** | 83.77 | 51.87 | 71.36 | 73. 3 |
| | 30 | 39.43 | 57.40 | 83.87 | 53.30 | 72.74 | 73.5 |
| | 50 | 39.31 | 57.27 | **84.14** | 53.85 | 73.63 | 74.6 |
| | 70 | 38.46 | 57.01 | 84.10 | **54.06** | **73.94** | 75.5 |
| QA 2-document | 0 | 22.75 | 37.25 | 87.16 | 42.93 | 56.73 | 149.8 |
| | 10 | 23.41 | 37.72 | 87.33 | 43.18 | 56.50 | 149.4 |
| | 30 | 23.02 | 37.37 | 87.24 | 43.64 | 58.25 | 149.4 |
| | 50 | 23.65 | 38.33 | 87.61 | 43.98 | 59.21 | 152.2 |
| | 70 | **23.69** | **38.51** | **87.76** | **44.84** | **59.53** | 152.2 |
| QA 3-document | 0 | 21.05 | 35.04 | 87.81 | 40.32 | 51.37 | 156.6 |
| | 10 | 21.08 | **35.56** | 87.57 | 41.62 | 50.74 | 154.6 |
| | 30 | 21.39 | 35.49 | 87.78 | 40.96 | 51.33 | 155.9 |
| | 50 | **21.60** | 35.48 | 87.83 | **41.91** | **51.97** | 157.4 |
| | 70 | 21.10 | 35.11 | **87.95** | 41.39 | 51.58 | 158.9 |
| Hallucination modification | 0 | 32.35 | 53.04 | 80.49 | 65.07 | 80.85 | 64.8 |
| | 10 | 32.57 | 53.29 | 80.51 | 65.30 | **81.36** | 63.9 |
| | 30 | **33.72** | **53.98** | **80.69** | 64.53 | 80.91 | 63.6 |
| | 50 | 32.58 | 52.92 | 80.49 | 65.07 | 80.18 | 65.7 |
| | 70 | 31.77 | 52.13 | 80.12 | **65.80** | 81.06 | 66.9 |

Bold numbers indicate the best results.

similarity metrics, such as the bleu score. This indicates that in the hallucination dataset created by UHGEval, the hallucination information often pertains to only one sentence, which is a mistake at the word or entity level, and does not require the comprehension of long text. Hence, there is no need to understand the whole document, only the relevant portions can be retrieved and modified.

### 4.3 Analyzing the Impact of Chunk Overlap on RAG Performance in Different Tasks

Chunk overlap is the number of tokens that two adjacent chunks share. To keep the text semantics coherent, adjacent chunks have some overlap. Chunk overlap determines the size of this overlap. This splitting method meets the maximum length limit of LLMs and maintains the semantic connection between adjacent chunks. Suitable chunk size and overlap can enhance the fluency and coherence of LLMs for long texts. We will show how the chunk overlap rate affects the system performance for different tasks in Table 4.

*Text Continuation.* With an increase in chunk overlap, we observe a slight enhancement in the metrics that evaluate the alignment of generated text with a reference answer (bleu, rouge-L, and bertScore). The RAGQuestEval metric, which evaluates the accuracy and completeness of important information, improves more obviously. These results indicate that a greater chunk overlap is beneficial for preserving the flow of ideas in the text, which is essential for tasks that require generating new, creative content.

*Open-Domain Multi-Document Summarization.* During summarization tasks, all evaluation metrics show a slight improvement as chunk overlap grows. Interestingly, despite assumptions that more overlap might reduce the variety of context information available, this does not result in a lower rate of recalling important information. In fact, the best performance in terms of recall occurs at a chunk overlap of 70%. This could mean that a larger overlap allows the model to focus more on the main points and ignore less relevant or redundant information.

*QA.* In QA tasks, chunk overlap has minimal impact on overall semantic similarity metrics such as bleu, rouge-L, and bertScore. However, it significantly affects the accuracy and recall metrics for key information. The results indicate that as chunk overlap increases, the accuracy and recall of key information in single-document QA tasks improve substantially. Similar improvements are observed in two-document QA tasks. However, for three-document QA tasks, the improvement is less pronounced. This may be because three-document QA tasks require richer context, and larger chunk overlaps may reduce the available context.

*Hallucination Modification.* Changes in chunk overlap have a minimal effect on the performance metrics for tasks that involve correcting hallucinations. This is likely due to the errors in these tasks typically being specific to individual entities or words, making the consistency of the chunks less impactful.

### 4.4 Analyzing the Impact of Retriever on RAG Performance in Different Tasks

A retriever is a key component of the RAG pipeline, which finds relevant documents from a large database based on the user input, and provides contextual information for the large model. There are two main types of retrievers: *keyword-based search-sparse retrieval algorithms*, which use keywords and their frequencies to compute the relevance between documents and queries. Common sparse retrieval algorithms include TF-IDF and BM25. BM25 is an enhanced TF-IDF method, which accounts for factors such as the length and position of words in the document. *Dense retrieval algorithms*, which use deep learning models to encode documents and queries into low-dimensional vectors, and then measure the cosine similarity between them. This method can capture the semantic and contextual information of words, and improve the retrieval performance.

In order to combine the advantages of both types of retrievers, we can fuse their retrieval results and randomly sample k from them as contexts for LLMs (*Hybrid*). Alternatively, we can also use a reranking model to rerank the fused retrieval results, and then select the top-k ones as the context of LLMs (*Hybrid+Rerank*). In our experiments, we employ the bge-rank as the rerank model.

*Text Continuation.* As Table 5 displays, the performance of the dense retriever is roughly equivalent to that of BM25, except for the key information recall rate. Compared to the keyword-based algorithm, the modern vector search can capture the semantic and contextual information of words, so that more content that does not match keywords but is obviously semantically related can be retrieved. However, the RAG system using BM25 also performs well. In terms of the precision of key information, BM25 even exceeds the dense retriever. This suggests that in the continuation task, which is a creative task, BM25 can retrieve content that is highly relevant to the user's intention, but may overlook some details.

Table 5. The Experimental Results for Evaluating Different Retrievers in Our Benchmark

| Task Name | Retriever Name | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|
| | | | | | Precision | Recall | |
| Text continuation | BM25 | 3.51 | 17.56 | 83.83 | **27.25** | 23.70 | 370.5 |
| | Dense | 3.66 | **17.78** | **83.99** | 26.96 | **24.68** | 367.6 |
| | Hybrid | **3.69** | 17.69 | 83.97 | 27.24 | 24.01 | 362.4 |
| | Hybrid+Rerank | 3.55 | 17.55 | 83.90 | 26.69 | 24.02 | 370.3 |
| Summarization | BM25 | **25.19** | 33.77 | 87.82 | **70.78** | 44.30 | 190.4 |
| | Dense | 23.69 | 33.53 | **88.49** | 68.06 | 46.18 | 205.9 |
| | Hybrid | 24.21 | 33.81 | 88.24 | 68.70 | 45.63 | 199.8 |
| | Hybrid+Rerank | 24.33 | **33.90** | 88.48 | 68.34 | **46.41** | 200.2 |
| QA 1-document | BM25 | 39.91 | 57.33 | 83.36 | 51.90 | 69.17 | 69.6 |
| | Dense | 39.76 | 57.24 | 83.81 | 52.67 | 70.82 | 73.3 |
| | Hybrid | 39.67 | 57.38 | 84.06 | 52.71 | 70.83 | 70.8 |
| | Hybrid+Rerank | **40.63** | **58.26** | **84.68** | **54.60** | **73.92** | 72.8 |
| QA 2-document | BM25 | **24.61** | 38.31 | 86.86 | 42.26 | 54.56 | 138.4 |
| | Dense | 22.75 | 37.25 | 87.16 | 42.93 | 56.73 | 149.8 |
| | Hybrid | 24.03 | 38.43 | 87.30 | 45.67 | 58.01 | 144.6 |
| | Hybrid+Rerank | 24.53 | **38.91** | **87.89** | **47.18** | **58.12** | 151.7 |
| QA 3-document | BM25 | 20.98 | 34.33 | 87.02 | 37.04 | 48.53 | 147.6 |
| | Dense | 21.05 | 35.04 | 87.81 | 40.32 | 51.37 | 156.6 |
| | Hybrid | 21.35 | 35.34 | 87.66 | 41.07 | 51.09 | 150.8 |
| | Hybrid+Rerank | **21.74** | **35.88** | **88.21** | **41.59** | **52.84** | 157.1 |
| Hallucination modification | BM25 | **33.09** | **54.21** | **80.86** | 64.80 | 79.90 | 59.0 |
| | Dense | 32.35 | 53.04 | 80.49 | 65.07 | 80.85 | 64.8 |
| | Hybrid | 32.22 | 52.92 | 80.57 | **66.30** | **81.03** | 63.4 |
| | Hybrid+Rerank | 32.62 | 53.01 | 80.62 | 65.57 | 80.82 | 64.9 |

Bold numbers indicate the best results.

*Open-Domain Multi-Document Summarization.* On the overall semantic similarity metric, the performance of the dense retriever is roughly equivalent to that of BM25. On the QuestEval metric, BM25 surpasses dense retriever in terms of key information precision, but slightly trails behind in key information recall. If the retrieved content contains a lot of irrelevant information, the model-generated summary may have errors or redundancies. BM25 retrieved content usually matches the user's intention better, but sometimes may miss some important information. Therefore, BM25 is weaker than dense retriever in key information recall, but excels in key information accuracy. Besides, hybrid retrieval algorithms presumably combine the advantages of both, and the RAG system generates content with suitable precision and recall.

*QA.* In QA, we find that dense retriever has a more obvious advantage over BM25, when dealing with reasoning questions that require synthesizing multiple documents. In QA tasks that require considering three documents, Dense retriever not only surpasses BM25 in all the overall semantic similarity metrics, but also achieves a significant improvement in key information precision and recall. This indicates that QA retrieval is more difficult than text continuation and other tasks, especially reasoning QA, which requires a higher level of semantic understanding, and simple keyword retrieval algorithms may not be sufficient. We also found that the Hybrid+Rerank algorithm,

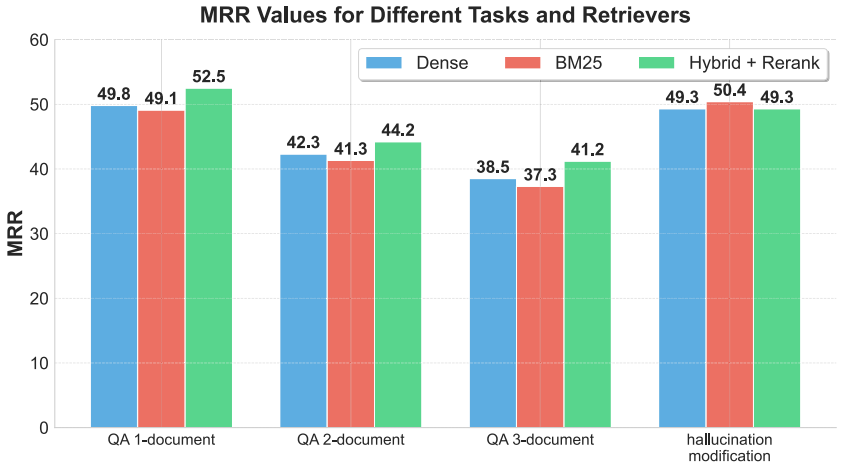


Fig. 7. Comparison of mean reciprocal rank (MRR) scores for different retrieval methods in our benchmark.

which combines and reranks the results of both algorithms, improves on all evaluation metrics. This suggests that this is a better retrieval algorithm for QA tasks.

*Hallucination Modification.* Consistent with the conclusion of summarization, the BM25 retriever performs slightly better than or equal to the dense retriever. For RAG tasks such as hallucination modification, which require precise retrieval of highly relevant content, BM25 shows good performance. Moreover, BM25 requires less computational resources than dense retrievers. This indicates that different RAG tasks require different retrieval algorithms.

*Retrieval Accuracy Evaluation.* To make a more comprehensive evaluation, we evaluated the retrieval accuracy on QA and hallucination modification tasks using **mean reciprocal rank (MRR)** as a separate metric. This separate evaluation allows for a more accurate assessment of the retriever's capabilities. Notably, text continuation and open-domain summarization tasks were excluded due to their subjective and vague evaluation criteria, lacking clear ground truth. Additionally, both 2-document and 3-document QA require multiple documents to address queries. Therefore, we calculate the MRR for each retrieved document individually and take the average as the final result. The pure hybrid algorithm was not evaluated separately as it could alter the order of retrieved content, affecting subsequent processing steps.

Figure 7 shows that the hybrid+reranking method excels in most tasks, outperforming other methods. This demonstrates the effectiveness of combining multiple retrieval strategies with reranking. Notably, BM25 and dense retrievers perform comparably in many cases, highlighting the strengths of both traditional and neural network methods. In QA, performance for all methods declines as the number of documents increases, aligning with expectations since multi-document tasks are more challenging and require stronger information integration. These results are consistent with our previous end-to-end evaluations, confirming the reliability of the end-to-end evaluation method.

### 4.5 Analyzing the Impact of Embedding Model on RAG Performance in Different Tasks

Most RAG systems use vector similarity-based algorithms as retrievers. Therefore, the embedding model that converts document blocks into vectors is crucial for the retrieval effect. We tested various embedding models optimized for retrieval tasks and compared their performance in the RAG system. We evaluated several embedding models with similar parameter sizes. According to

Table 6. The Experimental Results for Evaluating Different Embedding Models in Our Benchmark

| Task Name | Embedding Name | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|
| | | | | | Precision | Recall | |
| Text continuation | m3e-base | 3.59 | 17.55 | 83.76 | 27.30 | 23.73 | 350.0 |
| | bge-base | 3.66 | 17.78 | 83.99 | 26.96 | **24.68** | 367.6 |
| | stella-base | 3.73 | 17.67 | **84.05** | **28.78** | 24.65 | 366.6 |
| | gte-base | **3.76** | **17.80** | 84.03 | 27.35 | 24.18 | 362.1 |
| Summarization | m3e-base | 22.91 | 33.23 | 88.31 | 68.58 | 46.02 | 210.5 |
| | bge-base | 23.69 | **33.53** | 88.49 | 68.06 | 46.18 | 205.9 |
| | stella-base | **23.50** | 33.50 | 88.58 | **68.22** | 46.56 | 205.5 |
| | gte-base | 22.87 | 33.46 | **88.58** | 68.10 | **47.13** | 211.1 |
| QA 1-document | m3e-base | 38.81 | 56.49 | 83.41 | 50.18 | 69.72 | 75.2 |
| | bge-base | **39.76** | 57.24 | 83.81 | 52.67 | 70.82 | 73.3 |
| | stella-base | 39.58 | **57.28** | **83.91** | **53.13** | 71.74 | 73.9 |
| | gte-base | 39.58 | 57.19 | 83.90 | 52.39 | **71.97** | 76.5 |
| QA 2-document | m3e-base | 22.32 | 36.81 | 86.91 | 42.97 | 55.67 | 148.4 |
| | bge-base | 22.75 | 37.25 | 87.16 | 42.93 | 56.73 | 149.8 |
| | stella-base | **23.39** | **37.75** | 87.37 | **44.83** | **58.00** | 149.5 |
| | gte-base | 23.20 | 37.59 | **87.48** | 43.99 | 57.58 | 151.5 |
| QA 3-document | m3e-base | 20.72 | 34.78 | 87.43 | 39.57 | 50.88 | 154.3 |
| | bge-base | 21.05 | 35.04 | 87.81 | 40.32 | **51.37** | 156.6 |
| | stella-base | **21.26** | 35.27 | **87.81** | **41.4**1 | 50.42 | 154.4 |
| | gte-base | 21.15 | **35.59** | 87.86 | 40.18 | 51.11 | 157.2 |
| Hallucination modification | m3e-base | **32.83** | **53.27** | **80.78** | **65.87** | **81.69** | 64.5 |
| | bge-base | 32.35 | 53.04 | 80.49 | 65.07 | 80.85 | 64.8 |
| | stella-base | 32.34 | 52.96 | 80.59 | 65.74 | 81.50 | 65.2 |
| | gte-base | 31.69 | 52.46 | 80.40 | 65.35 | 80.69 | 64.5 |

Bold numbers indicate the best results.

[38], the embedding models' performance on the retrieval task should follow the order of GTE > STELLA > BGE > M3E. Our results in Table 6 show some variations with this order.

For creative tasks like continuation, the relevance of the retrieved content was often ambiguous. Thus, we noticed that the performance difference between the embedding models was small.

For single-document QA tasks that required precise localization of relevant documents, we found that m3e-base performed much worse than others. This matched the finding of [38]. However, for the hallucination modification task, m3e-base, which ranked the lowest on the retrieval benchmark, outperformed the other models on all metrics. These results further show that the retrieval benchmark may not be fully appropriate for RAG.

*Retrieval Accuracy Evaluation.* Similar to the experiments in the retriever evaluation, we use the MRR metric to evaluate four mainstream embedding methods: BGE, GTE, M3E, and STELLA. The results in Figure 8 indicate that the performance of these methods is relatively close across different tasks, with no single method outperforming the others in all tasks. This underscores the importance of considering specific task requirements when selecting an embedding method.

As the number of documents increases from 1 to 3, the MRR values for all methods show a downward trend. This trend aligns with our previous end-to-end experimental results, highlighting the challenges of multi-document understanding tasks.

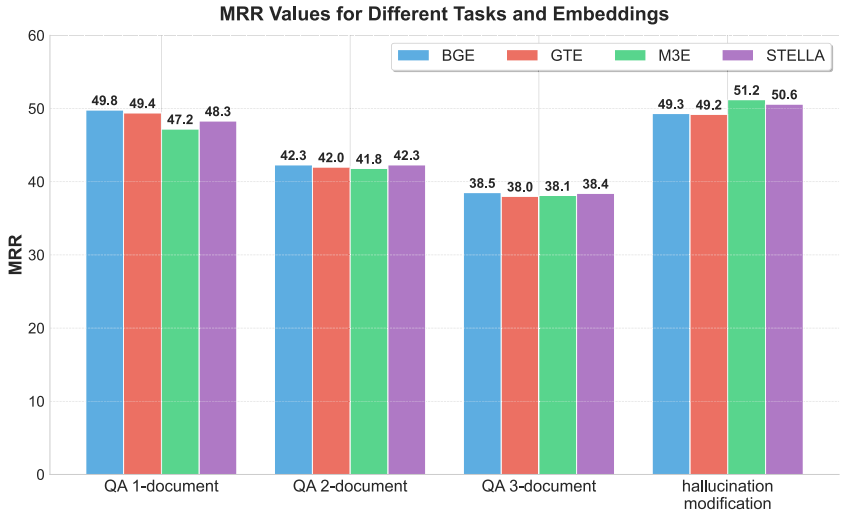


Fig. 8. Comparison of MRR scores for different embedding models in our benchmark.

### 4.6 Analyzing the Impact of Top-k on RAG Performance in Different Tasks

The RAG system converts the user's query into a vector using the same embedding model as the vector database. Then, it searches the index for the top-k most similar vectors to the query vector, and retrieves the corresponding text blocks from the database. These text blocks serve as the context for the LLM prompt. The amount of information that the model receives depends on the size of k. We will show how the amount of context information affects the system performance for different tasks in Table 7.

*Text Continuation.* Text continuation is a highly creative task. Table 7 shows that increasing top-k improves both the overall semantic similarity metrics (bertScore, bleu, and rouge-L) and the RAGQuestEval metrics. The recall metric of RAGQuestEval shows how much key information from the reference is included in the generated text, while the precision metric shows how correct and relevant that information is. We found that higher top-k values lead to higher recall and precision scores, indicating that the generated text contains more and better key information. We attribute this to the increased diversity and accuracy of the generated text from more documents.

*Open-Domain Multi-Document Summarization.* Increasing the top-k value leads to longer and lower-quality summaries. The rouge-L and bertScore metrics stay almost the same, but the bleu metric drops significantly, indicating less similarity between the summaries and the references. The top-k value also affects the key information metrics. Higher top-k values increase the recall scores, meaning more key information is included, but decrease the precision scores, meaning more errors or redundancies are present.

*QA.* For single-document QA, increasing top-k has little impact on the semantic similarity metric, but improves the RAGQuestEval metrics, which measure the accuracy and recall of key information. When the top-k value is too small, increasing the value of the top-k can significantly increase the recall and precision scores. This is because when the retrieved content is small, it may not be helpful for answering.

For multi-document QA (2-document and 3-document), increasing top-k significantly improves the recall and precision scores, as there are more chances to retrieve two relevant and complementary documents. More documents can also provide additional information, which helps to bridge

Table 7. The Experimental Results for Evaluating Different Top-k Values in Our Benchmark

| Task Name | Top-k | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|
| | | | | | Precision | Recall | |
| Text continuation | 2 | 2.89 | 17.20 | 83.60 | 25.35 | 23.14 | 367.0 |
| | 4 | 3.34 | 17.49 | 83.80 | 26.66 | 23.54 | 369.3 |
| | 6 | 3.53 | 17.64 | 83.81 | **27.66** | 24.32 | 375.4 |
| | 8 | 3.66 | 17.78 | 83.99 | 26.96 | 24.68 | 367.6 |
| | 10 | **3.91** | **17.84** | **84.01** | 27.61 | **25.00** | 355.7 |
| Summarization | 2 | **26.86** | **33.87** | 87.34 | **70.08** | 42.21 | 161.0 |
| | 4 | 24.78 | 33.62 | 87.95 | 68.91 | 44.19 | 185.6 |
| | 6 | 23.71 | 33.36 | 88.16 | 68.28 | 45.08 | 198.6 |
| | 8 | 23.69 | 33.53 | 88.49 | 68.06 | 46.18 | 205.9 |
| | 10 | 23.62 | 33.56 | **88.51** | 68.17 | **46.70** | 208.3 |
| QA 1-document | 2 | 39.13 | 56.26 | 82.57 | 50.81 | 65.80 | 67.7 |
| | 4 | 39.47 | 56.58 | 83.39 | 52.14 | 69.53 | 70.6 |
| | 6 | 39.40 | 56.86 | 83.81 | 52.60 | 70.80 | 72.5 |
| | 8 | **39.76** | **57.24** | 83.81 | 52.67 | **70.82** | 73.3 |
| | 10 | 38.84 | 56.52 | **83.93** | **53.67** | 70.31 | 74.1 |
| QA 2-document | 2 | 21.65 | 35.16 | 84.72 | 36.91 | 47.41 | 126.5 |
| | 4 | 22.33 | 36.68 | 86.39 | 41.15 | 52.78 | 139.5 |
| | 6 | **23.04** | 37.43 | 87.01 | 43.29 | 55.47 | 143.7 |
| | 8 | 22.75 | 37.25 | 87.16 | 42.93 | 56.73 | 149.8 |
| | 10 | 22.90 | **37.63** | **87.43** | **43.88** | **57.34** | 153.4 |
| QA 3-document | 2 | 19.27 | 32.57 | 85.65 | 33.70 | 43.90 | 136.3 |
| | 4 | 20.23 | 34.21 | 86.93 | 37.26 | 48.35 | 145.5 |
| | 6 | 20.73 | 34.95 | 87.66 | 39.59 | 51.03 | 151.3 |
| | 8 | **21.05** | **35.04** | 87.81 | 40.32 | 51.37 | 156.6 |
| | 10 | 20.61 | 35.01 | **88.02** | **40.90** | **52.11** | 162.5 |
| Hallucination modification | 2 | 32.12 | 53.00 | **80.54** | 64.95 | 79.24 | 59.6 |
| | 4 | **32.50** | 52.94 | 80.53 | **65.18** | 79.34 | 60.2 |
| | 6 | 32.32 | 52.70 | 80.36 | 64.48 | 79.27 | 61.8 |
| | 8 | 32.35 | **53.04** | 80.49 | 65.07 | 80.85 | 64.8 |
| | 10 | 31.30 | 51.71 | 80.09 | 64.84 | **80.90** | 68.3 |

Bold numbers indicate the best results.

the knowledge gap between documents and give more comprehensive answers. The results of 2-document and 3-document QA are similar.

*Hallucination Modification.* The top-k value has little effect on the semantic similarity metrics (bleu, rouge and bertScore) and the key information metric (RAGQuestEval). They only drop sharply when the top-k is too large. This is because, in our hallucination modification dataset, correcting the wrong information only requires a small amount of context, and the model has a certain anti-interference ability in the hallucination modification task, so the top-k value is not a decisive factor.

### 4.7 Analyzing the Impact of LLM on RAG Performance in Different Tasks

The core of the RAG system is an LLM, which can generate accurate and fluent answers based on the user's question and the retrieved information. In this article, we conducted experiments on several commonly used LLMs, as Table 8 displayed.

*Text Continuation.* The experimental results show that the larger the model parameters, the better the performance. GPT-4 surpassed other large models in all tasks, demonstrating its powerful generation ability.

*Open-Domain Multi-Document Summarization.* GPT-4 also excelled in the summary generation task. It achieved higher scores than other models on the overall semantic accuracy metric, as well as the key information recall and precision metric. Moreover, the summary generated by GPT-4 was relatively concise, avoiding redundant information. GPT-4 is the most suitable model for this task.

*QA.* For single-document QA, which only requires extracting relevant information from a sentence in the text, this task is relatively simple. Qwen and Baichuan2 even outperformed GPT series models. However, for multi-document QA that requires a comprehensive understanding of multiple documents, GPT-4 was far ahead of other models, showing its excellent knowledge fusion ability. The Baichuan2-13B model also performed better than GPT-3.5, indicating its potential.

*Hallucination Modification.* We found that some models generated text that was too long, introducing redundant information. The hallucination modification task only requires modifying the hallucination information, retaining other information, and not introducing irrelevant information. Therefore, ChatGLM2, Qwen-7B, and Baichuan2 did not complete this task well.

In summary, the GPT-4 model performed excellently on most tasks and evaluation metrics, proving that it is a powerful LLM. Qwen-7B and Qwen-14B models also performed well, especially in the text continuation and summary generation tasks. Baichuan2-13B model was very competitive with GPT-4 in the QA task, deserving more investigation.

*Latest LLM Evaluation.* Our dataset was constructed in December 2023. To evaluate its challenge to the latest LLMs in 2024, we experimented with two newly released models: GPT-4o (Released in May 2024) and Qwen2-7b (Released in June 2024).

The results show that GPT-4o performs similarly to its predecessor GPT-4, or with some slight improvements. In contrast, Qwen2-7b demonstrates improvements over its predecessor Qwen-7b in some tasks. These findings confirm that our benchmarks remain challenging for the latest LLMs. Additionally, it is encouraging to observe that the performance of many LLMs continues to improve with each new version.

### 4.8 Suggestions for Optimizing Your RAG System

Using the benchmark we constructed, we systematically evaluated the impact of each component in the RAG system in various application scenarios. Subsequently, we offer some suggestions for future researchers aiming to optimize the performance of the RAG system. Table 9 summarizes our recommendations.

The *top-k* value is a crucial parameter for the RAG system, as it determines how many documents are retrieved for each query. Depending on the scenario, the optimal top-k value may vary. For instance, in creative content generation tasks, such as text continuation, a larger top-k value is preferable. This allows the LLMs to access more diverse and relevant knowledge, resulting in richer and more accurate content. However, this also comes with a higher computational cost. In summary tasks, a moderate top-k value can strike a balance between precision and recall of information. For scenarios that require high precision, a smaller top-k value is recommended, while for scenarios that require high recall, a larger top-k value is recommended. In single-document QA, it is still recommended to use a large top-k value, which means that the answer can be

Table 8. The Experimental Results for Evaluating Different LLMs in Our Benchmark

| Task Name | Model Name | bleu | rouge-L | bertScore | RAGQuestEval | | Length |
|---|---|---|---|---|---|---|---|
| | | | | | Precision | Recall | |
| Text continuation | ChatGLM2-6B | 2.06 | 13.35 | 68.51 | 20.68 | 15.44 | 363.3 |
| | Qwen-7B | **7.10** | 15.31 | 77.94 | 28.06 | 18.44 | 159.6 |
| | Baichuan2-13B | 3.97 | 14.21 | 71.75 | 28.62 | 22.95 | 358.4 |
| | Qwen-14B | 5.70 | 18.48 | 82.97 | 27.89 | 21.68 | 240.1 |
| | GPT-3.5-turbo | 3.66 | 17.78 | 83.99 | 26.96 | 24.68 | 367.6 |
| | GPT-4-0613 | 5.58 | **19.47** | **84.91** | 30.34 | **28.02** | 369.8 |
| | Qwen2-7B | 2.94 | 16.76 | 83.82 | 26.90 | 23.68 | 350.0 |
| | GPT-4o | 4.48 | 18.85 | 84.45 | **30.89** | 26.11 | 356.7 |
| Summarization | ChatGLM2-6B | 17.09 | 28.16 | 83.00 | 58.94 | 40.35 | 228.1 |
| | Qwen-7B | 28.30 | 30.21 | 84.26 | 67.62 | 40.03 | 240.5 |
| | Baichuan2-13B | 24.49 | 32.49 | 85.64 | 65.96 | 42.53 | 179.5 |
| | Qwen-14B | **32.51** | 33.33 | 85.62 | 68.94 | 40.57 | 139.1 |
| | GPT-3.5-turbo | 23.69 | 33.53 | 88.49 | 68.06 | 46.18 | 205.9 |
| | GPT-4-0613 | 24.54 | **35.91** | 89.39 | **71.24** | 50.53 | 194.6 |
| | Qwen2-7B | 14.82 | 30.00 | 88.60 | 62.04 | 45.93 | 283.2 |
| | GPT-4o | 23.24 | 35.40 | **89.65** | 68.28 | **50.93** | 217.7 |
| QA 1-document | ChatGLM2-6B | 29.11 | 47.57 | 79.59 | 50.06 | 69.35 | 90.8 |
| | Qwen-7B | 39.63 | 56.71 | 82.64 | 51.77 | 72.02 | 68.8 |
| | Baichuan2-13B | 35.40 | 53.85 | 83.59 | 54.35 | **76.92** | 91.3 |
| | Qwen-14B | 37.95 | 55.13 | 83.25 | 53.03 | 73.92 | 73.8 |
| | GPT-3.5-turbo | **39.76** | **57.24** | **83.81** | 52.67 | 70.82 | 73.3 |
| | GPT-4-0613 | 33.87 | 51.42 | 80.92 | 53.14 | 62.39 | 95.9 |
| | Qwen2-7B | 23.06 | 41.25 | 82.10 | 60.07 | 72.17 | 123.3 |
| | GPT-4o | 33.32 | 51.78 | 83.35 | **65.33** | 66.59 | 74.7 |
| QA 2-document | ChatGLM2-6B | 15.15 | 29.12 | 82.30 | 37.61 | 51.51 | 193.4 |
| | Qwen-7B | 22.61 | 36.07 | 85.84 | 42.32 | 56.26 | 157.6 |
| | Baichuan2-13B | 20.32 | 35.56 | 87.49 | 45.01 | 61.47 | 208.8 |
| | Qwen-14B | 21.11 | 34.97 | 85.87 | 42.23 | 56.59 | 151.1 |
| | GPT-3.5-turbo | 22.75 | **37.25** | 87.16 | 42.93 | 56.73 | 149.8 |
| | GPT-4-0613 | 20.38 | 36.08 | 88.10 | **49.56** | 62.56 | 223.0 |
| | Qwen2-7B | 15.26 | 41.25 | 82.10 | 48.89 | 61.41 | 209.1 |
| | GPT-4o | **22.84** | 36.61 | **88.38** | 44.04 | **67.44** | 124.3 |
| QA 3-document | ChatGLM2-6B | 14.01 | 27.71 | 83.42 | 35.60 | 45.28 | 204.1 |
| | Qwen-7B | 21.63 | 33.42 | 86.31 | 39.14 | 50.55 | 160.6 |
| | Baichuan2-13B | 18.30 | 33.34 | 88.08 | 41.35 | 55.75 | 227.5 |
| | Qwen-14B | 19.83 | 33.33 | 86.93 | 42.01 | 51.70 | 161.2 |
| | GPT-3.5-turbo | 21.05 | 35.04 | 87.81 | 40.32 | 51.37 | 156.6 |
| | GPT-4-0613 | 19.11 | 34.58 | 88.88 | **48.24** | 56.48 | 235.1 |
| | Qwen2-7B | 16.23 | 32.18 | 87.69 | 45.72 | 55.29 | 207.2 |
| | GPT-4o | **22.84** | **35.98** | **89.21** | 43.56 | **63.90** | 139.9 |
| Hallucination modification | ChatGLM2-6B | 13.51 | 28.70 | 71.26 | 59.63 | 73.02 | 176.0 |
| | Qwen-7B | 22.87 | 38.10 | 73.52 | 60.00 | 73.72 | 172.5 |
| | Baichuan2-13B | 10.56 | 27.28 | 68.90 | 54.42 | 67.47 | 124.8 |
| | Qwen-14B | 33.78 | 51.90 | 79.49 | 67.05 | **84.08** | 89.7 |
| | GPT-3.5-turbo | 32.35 | 53.04 | 80.49 | 65.07 | 80.85 | 64.8 |
| | GPT-4-0613 | **36.69** | **55.70** | **81.27** | **69.18** | 82.06 | 63.5 |
| | Qwen2-7B | 31.07 | 52.91 | 80.25 | 65.48 | 79.16 | 49.3 |
| | GPT-4o | 36.73 | 54.79 | 80.90 | 63.61 | 73.75 | 51.9 |

Bold numbers indicate the best results.

Table 9. Recommendations for Adjusting RAG System Key Parameters Based on Different Tasks

| Scenario | Top-k | Chunk Size | Chunk Overlap | Retriever | LLM |
|---|---|---|---|---|---|
| Create: Creative content generation | Larger, to access diverse knowledge | Larger, to preserve article structure | Larger, to maintain semantic coherence | Dense algorithm for semantic understanding | Qwen-14B for cost-effective high-quality text |
| Delete: Summarization | Moderate, for precision-recall balance | Smaller for more recall, larger for more precision | Larger, to maintain semantic coherence | BM25 for precise content, Dense algorithm for more recall | Qwen-14B for high-quality summaries |
| Read: Single-document QA | Larger, for repeated determination | Moderate, for pinpointing short answers | Larger, to maintain semantic coherence | Hybrid+rerank for enhanced performance | Baichuan2-13B for GPT-4-like performance |
| Read: Multi-document QA | Larger, for retrieving complementary articles | Larger, for article completeness | Larger, to maintain semantic coherence | Hybrid+rerank for enhanced performance | Baichuan2-13B for GPT-4-like performance |
| Update: Error correction | Smaller, for high precision tasks | Larger, to avoid breaking article structure | Smaller, error correction tasks are not sensitive to semantic coherence | BM25 for precise content generation | GPT-4 or alternatives depending on cost |

determined multiple times. In QA tasks that involve reasoning across multiple documents, a larger top-k value can help to retrieve two related and complementary articles, thus enhancing the QA performance.

The *chunk size* is also an important factor when building the vector index for external knowledge. For creative scenarios, such as content generation, we suggest using a larger chunk size to preserve the structure of the article and avoid affecting the performance of the RAG system. For summary scenarios, a smaller chunk size can be used if more information is desired to be recalled; however, if the precision of the generated content is more important, a larger chunk size is still recommended to avoid destroying the structure of the article. In factual QA scenarios, a smaller chunk size is beneficial for finding the answer in a short sentence. For reasoning tasks, a larger chunk size can ensure the article's completeness and enhance the reasoning ability.

The *chunk overlap* is the shared content between two adjacent chunks, and chunk overlap is key to maintaining the coherence of semantics in LLMs when dealing with long texts. Experiments show that for creative generation, summarization, and QA scenarios, the semantic consistency between chunks is very important, so a large chunk overlap value should be maintained. However, for error correction scenarios, the semantic consistency between chunks is relatively unimportant, and a smaller chunk overlap value can be considered.

When choosing an *embedding* model, you can refer to the mteb leaderboard [38], which shows the performance of different embedding models on retrieval tasks. However, the actual performance of the RAG system may differ from the leaderboard, so you need to evaluate and adjust according to the specific scenario.

When choosing a *retrieval algorithm*, BM25 has the advantage of saving computational resources compared to dense retrievers, and since it is a keyword-based algorithm, it can usually retrieve very relevant documents. However, keyword-based algorithms perform poorly in capturing semantics

and may miss some relevant content. Therefore, we suggest using BM25 for tasks that require precise content generation, such as hallucination modification and summarization.

However, BM25 may not be suitable for tasks that require semantic understanding, such as QA and creative generation, and we recommend using dense algorithms based on deep learning embeddings instead.

Moreover, the hybrid algorithm that combines dense and BM25 retriever has very limited improvement on the overall quality of the generated results. However, by using a rerank model to reorder the retrieval results and then inputting them into LLMs, the performance of almost all tasks improved, especially reasoning tasks. Therefore, we suggest trying to use the hybrid algorithm+rerank retrieval mode when the conditions permit, which can achieve better performance in the RAG system.

When choosing an *LLM*, GPT-4 model is undoubtedly the most advanced model at present. However, due to the high cost of invoking GPT-4, we may need to consider some open-source alternatives. According to our experimental results, Qwen-14B model has shown similar performance to GPT-4 in the two tasks of text continuation and summary generation, and can generate high-quality creative and summarizing texts. In the QA task, Baichuan2-13B model also showed a level close to GPT-4, and can generate accurate and fluent answers. Therefore, we can choose a suitable LLM according to different tasks and cost requirements.

## 5 Conclusion

In this article, we have introduced an innovative framework (CRUD-RAG) for evaluating RAGsystems that is both comprehensive and scenario-specific. Our unique categorization of text generation tasks into the CRUD—Create, Read, Update, and Delete—types provides a structured approach to assess the capabilities and limitations of RAG systems in handling a variety of textual contexts. To facilitate this evaluation, we have meticulously constructed large-scale datasets for each CRUD category, which are tailored to challenge and reflect the performance of RAG systems under different operational conditions. Through rigorous experimental comparisons, we have demonstrated that RAG systems can significantly enhance the quality of generated content by effectively incorporating information from external knowledge sources.

Our study delves into the intricate balance required in the fine-tuning process of RAG systems, highlighting the importance of optimizing the retrieval model, context length, construction of the knowledge base, and the deployment of the underlying LLM to achieve the best results. The insights provided by our findings offer a valuable roadmap for researchers and practitioners in the field, guiding them in the development and refinement of RAG systems. We believe that the methodologies and results presented in this article will spur further exploration and innovation in the realm of RAG technologies. Our work aims to catalyze advancements in text generation applications, pushing the envelope of what is possible with the integration of retrieval mechanisms and language models. We hope that this contribution will serve as a cornerstone for future research efforts, fostering the creation of more intelligent, adaptive, and context-aware generative systems.

## Acknowledgments

The authors would like to extend our special thanks to five colleagues from Xinhua News Agency: Luo Yi, Cheng Peng, Deng Haiying, Wang Zhonghao, and Lu Zijia. Their expertise in news data was invaluable; they provided important news data for our dataset and helped guide us in manually adjusting the dataset.

## References


[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 475–484.

[2] Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts (ACL '23)*. 41–46.

[3] Garbiel Bénédict, Ruqing Zhang, and Donald Metzler. 2023. GEN-IR@ SIGIR 2023: The first workshop on generative information retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3460–3463.

[4] Alec Berntson. 2023. Azure AI Search: Outperforming Vector Search with Hybrid Retrieval and Ranking Capabilities. Retrieved from https://techcommunity.microsoft.com/t5/ai-azure-ai-services-blog/azure-ai-search-outperforming-vector-search-with-hybrid/ba-p/3929167

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv:2303.12712. Retrieved from https://arxiv.org/abs/2303.12712

[6] Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. 6251–6258.

[7] Shuyang Cao and Lu Wang. 2024. Verifiable generation with subsentence-level fine-grained citations. arXiv:2406.06125.

[8] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Benchmarking large language models in retrieval-augmented generation. arXiv:2309.01431. Retrieved from https://arxiv.org/abs/2309.01431

[9] Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 245–255.

[10] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift Yourself Up: Retrieval-augmented text generation with self memory. arXiv:2305.02437. Retrieved from https://arxiv.org/abs/2305.02437

[11] Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *Proceedings of the 11th International Conference on Learning Representations (ICLR '23)*.

[12] Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. arXiv:2005.03754. Retrieved from https://arxiv.org/abs/2005.03754

[13] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAs: Automated evaluation of retrieval augmented generation. arXiv:2309.15217. Retrieved from https://arxiv.org/abs/2309.15217

[14] Joe Ferrara, Ethan-Tonic, and Oguzhan Mete Ozturk. 2024. The RAG Triad. Retrieved from https://www.trulens.org/trulens_eval/core_concepts_rag_triad/

[15] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J. F. Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 795–798.

[16] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '23)*, 1762–1777.

[17] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*, 6465–6488.

[18] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997. Retrieved from https://arxiv.org/abs/2312.10997

[19] Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. arXiv:2301.00303. Retrieved from https://arxiv.org/abs/2301.00303

[20] Ivan Ilin. 2023. Advanced RAG Techniques: An Illustrated Overview. Retrieved from https://pub.towardsai.net/advanced-rag-techniques-an-illustrated-overview-04d193d8fec6

[21] Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. arXiv:2208.03299. Retrieved from https://arxiv.org/abs/2208.03299

[22] Bin Ji, Huijun Liu, Mingzhe Du, and See-Kiong Ng. 2024. Chain-of-thought improves text generation with citations in large language models. In *Proceedings of 38th AAAI Conference on Artificial Intelligence (AAAI '24), 36th Conference on Innovative Applications of Artificial Intelligence (IAAI '24), 14th Symposium on Educational Advances in Artificial Intelligence (EAAI '14)*, 18345–18353.

[23] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*, 7969–7992.

[24] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. arXiv:2305.18846. Retrieved from https://arxiv.org/abs/2305.18846

[25] Haim Kilov. 1990. From semantic to object-oriented data modeling. In *Proceedings of the 1st International Conference on Systems Integration (Systems Integration '90)*. IEEE, 385–393.

[26] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[27] Langchain. 2023. Evaluating RAG Architectures on Benchmark Tasks. Retrieved from https://langchain-ai.github.io/langchain-benchmarks/notebooks/retrieval/comparing_techniques.html

[28] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS '20)*.

[29] Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. 2024. Citation-enhanced generation for LLM-based chatbots. arXiv:2402.16063. Retrieved from https://arxiv.org/abs/2402.16063

[30] Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023. Are ChatGPT and GPT-4 general-purpose solvers for financial text analytics? A study on several typical tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 408–422.

[31] Xun Liang, Shichao Song, Simin Niu, Zhiyu Li, Feiyu Xiong, Bo Tang, Zhaohui Wy, Dawei He, Peng Cheng, Zhonghao Wang, et al. 2023. UHGEval: Benchmarking the hallucination of Chinese large language models via unconstrained generation. arXiv:2311.15296. Retrieved from https://arxiv.org/abs/2311.15296

[32] Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics (EMNLP '23)*, 7001–7025.

[33] Qi Liu, Gang Guo, Jiaxin Mao, Zhicheng Dou, Ji-Rong Wen, Hao Jiang, Xinyu Zhang, and Zhao Cao. 2024. An analysis on matching mechanisms and token pruning for late-interaction models. *ACM Transactions on Information Systems* 42, 5 (2024), 118:1–118:28.

[34] Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. arXiv:1905.13164. Retrieved from https://arxiv.org/abs/1905.13164

[35] Yuanjie Lyu, Chen Zhu, Tong Xu, Zikai Yin, and Enhong Chen. 2022. Faithful abstractive summarization via fact-aware consistency-constrained transformer. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1410–1419.

[36] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. arXiv:2305.14283. Retrieved from https://arxiv.org/abs/2305.14283

[37] Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples!. In *Findings of the Association for Computational Linguistics (EMNLP '23)*, 10572–10601.

[38] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. arXiv:2210.07316. Retrieved from https://arxiv.org/abs/2210.07316

[39] Darren Oberst. 2023. How to Evaluate LLMs for RAG? Retrieved from https://medium.com/@darrenoberst/how-accurate-is-rag-8f0706281fd9

[40] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. KILT: A benchmark for knowledge intensive language tasks. arXiv:2009.02252. Retrieved from https://arxiv.org/abs/2009.02252

[41] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 539–548.

[42] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics* 49, 4 (2023), 777–840.

[43] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '21)*, 1172–1183.

[44] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. ARES: An automated evaluation framework for retrieval-augmented generation systems. arXiv:2311.09476. Retrieved from https://arxiv.org/abs/2311.09476

[45] Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. QuestEval: Summarization asks for fact-based evaluation. arXiv:2103.12693. Retrieved from https://arxiv.org/abs/2103.12693

[46] Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In ChatGPT we trust? Measuring and characterizing the reliability of ChatGPT. arXiv:2304.08979. Retrieved from https://arxiv.org/abs/2304.08979

[47] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. REPLUG: Retrieval-augmented black-box language models. arXiv:2301.12652. Retrieved from https://arxiv.org/abs/2301.12652

[48] Ciprian-Octavian Truica, Florin Radulescu, Alexandru Boicea, and Ion Bucur. 2015. Performance evaluation for CRUD operations in asynchronously replicated document oriented database. In *Proceedings of the 20th International Conference on Control Systems and Computer Science*. IEEE, 191–196.

[49] Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. arXiv:2004.04228. Retrieved from https://arxiv.org/abs/2004.04228

[50] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*, 9414–9423.

[51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 35, 24824–24837.

[52] Muning Wen, Runji Lin, Hanjing Wang, Yaodong Yang, Ying Wen, Luo Mai, Jun Wang, Haifeng Zhang, and Weinan Zhang. 2023. Large sequence models for sequential decision-making: A survey. *Frontiers of Computer Science* 17, 6 (2023), 176349.

[53] Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. MindMap: Knowledge graph prompting sparks graph of thoughts in large language models. arXiv:2308.09729. Retrieved from https://arxiv.org/abs/2308.09729

[54] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*. Retrieved from https://journal.hep.com.cn/fcs/EN/10.1007/s11704-024-40555-y

[55] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: Improving retrieval-augmented LMs with compression and selective augmentation. arXiv:2310.04408. Retrieved from https://arxiv.org/abs/2310.04408

[56] Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2024. ALiiCE: Evaluating positional fine-grained citation generation. arXiv:2406.13375. Retrieved from https://arxiv.org/abs/2406.13375

[57] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 5364–5375.

[58] Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*, 7170–7186.

[59] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1933–1936.

[60] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-Note: Enhancing robustness in retrieval-augmented language models. arXiv:2311.09210. Retrieved from https://arxiv.org/abs/2311.09210

[61] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems* 22, 2 (2004), 179–214.

[62] Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024. NoteLLM: A retrievable large language model for note recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, 170–179.

[63] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. arXiv:2310.07554. Retrieved from https://arxiv.org/abs/2310.07554

[64] Weijia Zhang, Mohammad Aliannejadi, Yifei Yuan, Jiahuan Pei, Jia-Hong Huang, and Evangelos Kanoulas. 2024. Towards fine-grained citation evaluation in generated text: A comparative analysis of faithfulness metrics. arXiv:2406.15264. Retrieved from https://arxiv.org/abs/2406.15264

[65] Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. ChatGPT hallucinates when attributing answers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 46–51.

Received 3 February 2024; revised 3 July 2024; accepted 6 October 2024