Ragnarök: A Reusable RAG Framework and Baselines for TREC 2024 Retrieval-Augmented Generation Track

Ronak Pradeep[1](✉), Nandan Thakur[1], Sahel Sharifymoghaddam[1], Eric Zhang[1], Ryan Nguyen[1], Daniel Campos[2], Nick Craswell[3], and Jimmy Lin[1]

[1] University of Waterloo, Waterloo, Canada
rpradeep@uwaterloo.ca
[2] Snowflake Inc., New York, USA
[3] Microsoft, Seattle, USA

Abstract. Have you tried the new Bing Search? Or maybe you fiddled around with Google AI Overviews? These might sound familiar because the modern-day search stack has evolved to include retrieval-augmented generation (RAG) systems. They allow searching and incorporating real-time data into large language models (LLMs) to provide a well-informed, attributed, concise summary, in contrast to the traditional search paradigm that relies on displaying a ranked list of documents. Therefore, given these recent advancements, it is crucial to have an arena to build, test, visualize, and systematically evaluate RAG-based search systems. With this in mind, we propose TREC RAG to foster innovation in evaluating RAG systems. In our work, we lay out the steps we have taken towards making this track a reality—we describe the details of our reusable framework, Ragnarök, explain the curation of the new MS MARCO V2.1 collection, release the development topics, some relevance judgments and baselines for the track and standardize the I/O definitions which assist the end user. Next, using Ragnarök, we identify and provide key proprietary and open-source baselines such as OpenAI's GPT-4o, Cohere's Command R+, and Meta's LLaMA3.1-70B. Further, we introduce a web-based user interface for an interactive arena allowing benchmarking pairwise RAG systems by crowdsourcing. We open-source Ragnarök and baselines to achieve a unified standard for future RAG systems.

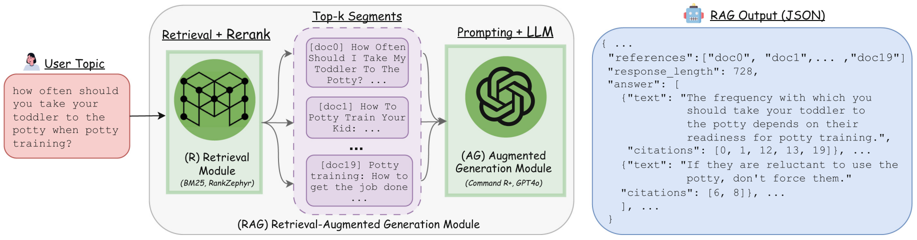
Keywords: Retrieval-Augmented Generation · Large Language Models · Ad Hoc Retrieval

1 Introduction

Retrieval-Augmented Generation (RAG) [9,21,23,33] has emerged as a popular technique to augment large language model (LLM) generation for knowledge-intensive tasks such as open-domain question answering or fact verification [51].

R. Pradeep and N. Thakur—Both authors contributed equally to this research.

Using the top-k retrieved segments from a suitable retrieval system, RAG systems output an answer summary grounded on the relevant context. RAG systems mitigate factual inconsistencies in LLM outputs [19,27,33,39], and enhance interpretability [21] and generalization [20], thus facilitating a wider LLM adoption across several domains like Medicine [63] and Finance [24].



```
how often should
you take your
toddler to the
potty when potty
training?
```

```
[doc0] How Often
Should I Take My
Toddler To The
Potty? ...

[doc1] How To
Potty Train Your
Kid: ...

...

[doc19] Potty
training: How to
get the job done
...
```

```
{ ...
 "references":["doc0", "doc1",... ,"doc19"]
 "response_length": 728,
 "answer": [
   {"text": "The frequency with which you
            should take your toddler to
            the potty depends on their
            readiness for potty training.",
    "citations": [0, 1, 12, 13, 19]}, ...
   {"text": "If they are reluctant to use the
            potty, don't force them."
   "citations": [6, 8]}, ...
   ], ...
 }
```

Fig. 1. Schematic diagram of the Ragnarök. Given a user topic (left), the process consists of two steps: (1) (R) retrieval (+ rerank), where the topic yields the top-k relevant segments from our document collection (e.g., potty training articles); and (2) (AG) augmented-generation, where the retrieved segments with a suitable prompt template are fed to the large language model (LLM) to generate the post-processed answer response (JSON) containing individual sentence-level citations.

Several companies provide end-to-end RAG systems such as Bing Search [45], or Google AI Overviews. Most of these systems are either proprietary or offer limited user customization. Likewise, the absence of a standardized framework makes implementing RAG at a large scale challenging. Implementing atop existing frameworks requires custom code for multiple steps including retrieval, reranking, and generation. To promote wider adoption of RAG in academia, we develop Ragnarök, a user-friendly, reusable, end-to-end RAG framework offering code to customize retrievers, rerankers, and generation models.

Ragnarök comprises two key modules: (R) Retrieval and (AG) Augmented Generation. The retrieval module incorporates both the retrieval and reranking stages to yield the top-k segments for a given user topic. Next, the augmented generation module uses both the user topic and retrieved segments as input to produce a RAG answer, formatted into individual sentences, citing the relevant information from the top-k retrieved and reranked segments. Ragnarök is deeply integrated with existing Python frameworks, such as `Pyserini` [36] and `rank_llm` [53,54] and can be easily installed via PyPI using "`pip install pyragnarok`". This framework offers REST APIs and an integrated WebUI to enhance user-friendliness and improve the human evaluation experience.

Ragnarök is used to provide baselines in the TREC 2024 Retrieval-Augmented Generation Track. An ideal framework should contain a sufficiently large document collection covering diverse information and non-factoid, decompositional topics requiring long-form answers. In our work, we deduplicate the existing MS MARCO V2 document collection. In addition, we provide a "segment" collection using a sliding-window chunking technique (discussed more in

Sect. 4). Further, we release two sets of development topics: (i) TREC-RAGgy 2024: a filtered subset of topics with long-form answers from TREC Deep Learning 2021–23 [14–16]; and (ii) TREC-Researchy 2024: a subset of the Researchy Questions introduced in Rosset et al. [57].

Ragnarök supports a head-to-head RAG battle arena for answer evaluation, heavily inspired by recent work such as the Chatbot Arena [12,66]. We include key proprietary and open-source baselines such as OpenAI GPT-4o [48], Cohere Command R+ [13], and Meta LLaMA3.1-70B [17] and evaluate some of the baselines using the retrieval setup involving BM25 [55] and RankZephyr [54] with human preferences. Overall, we observe GPT-4o to provide more detailed answers over Command R+ on the development set of topics (discussed more in Sect. 6).

Ragnarök already supports the TREC 2024 Biomedical Generative Retrieval (BioGen) Track, as well as report generation for the TREC 2024 Neural Cross-Language Information Retrieval (NeuCLIR) Track. This demonstrates the framework's adaptability across diverse domains, each characterized by distinct information needs. The flexibility and domain-general capabilities highlight Ragnarök and its potential to serve as a versatile tool in specialized RAG applications.

Finally, with the growing need for standardized RAG frameworks, we have made Ragnarök publicly available.[1] Moving forward, we plan to expand our retrieval corpora, incorporate additional baseline systems, and continuously enhance our framework to meet our new goals.

2 Related Work

RAG Frameworks. Existing RAG systems are primarily proprietary and difficult to reproduce. Open-source frameworks such as LangChain [11] and LlamaIndex [38], while available, are not research-friendly and lack proper evaluation and benchmarking. FlashRAG [25], a concurrent work, is a similarly motivated toolkit to improve the RAG experience for researchers. While the framework is extensive and designed for pipeline flexibility, Ragnarök offers a few additional capabilities—a WebUI serving a RAG battle arena, REST APIs, a standardized I/O definition working with sentence-level citations, and a tight integration with popular retrieval and reranking frameworks like Pyserini [36] and RankLLM.

Collection Selection. Current RAG datasets are constructed using the English Wikipedia as the document collection, However, their scale is limited to provide rich and comprehensive information to support RAG systems. A prominent option, ClueWeb22 [49] offers an extensive collection of 22 billion curated web pages, previously utilized in parts by TREC tracks such as the TREC Conversational Assistance Track (CAsT) [50] and the TREC Interactive Knowledge Assistance Track (iKAT) [3]. However, ClueWeb22 remains gated and serves as a significant hurdle for researchers with limited computational resources or those

[1] Ragnarök code repository: https://github.com/castorini/ragnarok.

unable to obtain access credentials. Another alternative is the MS MARCO V2 document collection, which was used primarily in the TREC Deep Learning (DL) track. The open availability of MS MARCO makes it an attractive option for researchers seeking to develop and evaluate RAG systems at scale.

Topic Selection. Recently, there has been a surge in datasets providing topics with long-form answers for evaluating RAG systems. ASQA [58], ELI5 [18], and QAMPARI [4] were utilized for evaluation in the Automatic LLMs' Citation Evaluation (ALCE) framework [19]. Similarly, related long-form QA datasets include AquaMuse [28], ExpertQA [42], and TruthfulQA [37]. Another recently introduced dataset is ClapNQ [56], created from the subset of Natural Questions (NQ) [29] and HAGRID [26] built on a subset of MS MARCO Dev [7]. Almost all previous datasets are built on English Wikipedia. In contrast, our work deliberately avoids English Wikipedia to prevent the overfitting seen in existing retrieval benchmarks [46,61]. In our work, we re-utilize topics from previous TREC tracks such as the Deep Learning (DL) track, because human judgments are available on the MS MARCO V2 corpora and Researchy Questions [57] as it covers a wide range of topics with multi-faceted information needs.

3 Our Framework

Ragnarök is an open-source, reproducible, and reusable framework implementing an end-to-end retrieval-augmented generation (RAG) pipeline, comprising two modules applied sequentially: (1) (R) retrieval and (2) (AG) augmented generation. Through Ragnarök, we provide several baselines to all participants in TREC RAG. An overview of the framework is provided in Fig. 1. We first describe both modules and expand on the I/O specifications in our framework.

Retrieval Module. This module retrieves the relevant segments for a user topic as the input. It supports (i) first-stage lexical retrieval models such as BM25 [55] and (ii) reranking models such as RankZephyr [54]. The retrieval system searches for relevant segments in the document collection and retrieves the top-100 segments further reranked by the reranker model to filter out the top-20 relevant segments for the next stage.

Augmented Generation Module. This module takes in the user topic and the top-20 retrieved segments (from the retrieval module) as the input and a prompting strategy to the LLM to generate the answer response with in-context citations for the topic. The answer response is divided into individual sentences, each sentence within the answer contains text and is grounded on retrieved documents provided as references to the LLM (if possible).

3.1 RAG Input/Output Definitions

RAG Input. The input specifications are straightforward as the user can formulate any question they wish to ask, provide the user topic as input, and call Ragnarök.

RAG Output. The user receives a JSON output in response to their topic from Ragnarök. The first key in the output JSON schema, `references`, provides a list of segment IDs that are referenced in the answer. These segments are selected from among the top-20 results returned by the retrieval module. Next, `answer`, provides the LLM-generated RAG answer to the user topic, presented as a top-to-bottom list of sentence-level texts with corresponding segment citations. All citations are zero-based indexed, indicating the exact position of the segment ID from the `references` list. Finally, `response_length`, provides the total count of the whitespace-separated words present in the output RAG answer.

Table 1. Comparison of document and segment counts between versions V2 and V2.1 (our version after removing near-duplicates) of the MS MARCO collection.

Collection	Version V2	Version V2.1 (Ours)
MS MARCO Document	11,959,635	10,960,555
MS MARCO Segment	124,131,414	113,520,750

4 Document Collection

The MS MARCO V2 document collection, previously used in the TREC-DL tracks, contains a substantial overlap of near-duplicates (documents with sufficiently similar text information) [15,16]. When left intact, these near-duplicates degrade the downstream retrieval accuracy and reduce the diversity of the collected documents, potentially impacting the effectiveness of RAG systems.

Documents in the existing collection tend to be verbose, containing extensive information about individual topics. Chunking, which breaks down a long verbose document into smaller compact representations is a key challenge, as the retrieved chunk representations correlate with the RAG answer quality [39].

MS MARCO V2.1 Document Collection. We conduct a deduplication strategy in the MS MARCO V2 document collection to avoid near-duplicates in two stages. In the first stage, we establish an equivalence class of the documents using Locality Sensitive Hashing (LSH) with MinHash [10] and 9-gram shingles. Next, we select a representative document for each equivalence class for our refined MS MARCO V2.1 document collection, reducing the duplicates in the original MS MARCO V2 document collection by 8.35% as shown in Table 1.

MS MARCO V2.1 Segment Collection. We segment the MS MARCO V2.1 document collection into overlapping segments (or chunks) and develop the MS MARCO V2.1 segment collection, with more than 113 million text segments (Table 1). We utilize a sliding window technique to generate the segments, by fixing the sliding window size of 10 sentences and a stride of 5 sentences to create each segment, roughly on average, between 500–1000 characters long. To easily map each segment back to the document, every segment contains the document ID within the segment ID. Further, two new fields: `start_char` and `end_char`, which indicate the start (inclusive) and the end position character (exclusive) of where the segment begins and ends in the mapped MS MARCO V2.1 document collection, respectively.

5 Topic Collection

Topics, i.e., user queries, are crucial for robust evaluation of RAG systems. Traditionally, popular retrieval and traditional QA benchmarks primarily consist of factoid queries, where answers are typically found within a single sentence or paragraph. However, these topics lack complexity, leading to short answers that can be easily memorized by LLMs. For instance, MS MARCO [7] surprisingly contains up to 55% factoid queries [8,57]. To avoid short-form answers in RAG, we utilize two collections containing non-factoid topics covering diverse information and requiring long-form answers. We describe these collections below:

Table 2. TREC-RAGgy 2024 and TREC-Researchy 2024 topic distribution. The table shows the top-5 categories in topic classification for TREC-RAGgy 2024 (each topic classified into a single category), intrinsic attributes for TREC-Researchy 2024, and the first word in all topics.

TREC-RAGgy 2024				TREC-Researchy 2024			
Topic Category	%	First Word	%	Intrinsic Attributes	%	First Word	%
Aggregation	24.2	What	37.5	Knowledge-Intensive	79.8	How	41.0
Simple w/ cond.	23.3	How	27.5	Multi-Faceted	75.7	Why	25.5
Set	20.8	Why	3.3	Reasoning-Intensive	75.5	What	15.0
Simple	10.0	Is	2.5	Subjective	48.5	Is	5.2
Comparison	6.7	When	1.7	Assumptive	25.7	Should	2.2

TREC-RAGgy 2024. We develop TREC-RAGgy 2024, a collection with topics filtered from TREC Deep Learning 2021–2023 tracks [14–16], based on topic category and generated-answer classification. We classify each available topic into seven categories and filter out a subset of topics that either have a long-form answer or require information aggregation across multiple sources of information. Out of the 210 original topics available, we filter and include 120 topics (57.1%) in the TREC-RAGgy 2024 collection. From Table 2, we observe 24.2% of the topics included are "aggregation", indicating RAG systems should aggregate information from multiple segments to generate an accurate long-form answer. Similarly, 65% of the topics start with "what" or "how" questions.

The TREC-RAGgy 2024 collection includes mapped document-level relevance judgments from previous TREC Deep Learning tracks. These relevance judgments, originally associated with the full documents, have been mapped to the new deduplicated corpus, enabling effective evaluation of the retrieval systems. However, since most of our work operates at the segment level, prior to the evaluation, we perform a MaxP step—leveraging the score of the most relevant segment as the representative score of the document.

TREC-Researchy 2024. Researchy Questions, introduced in Rosset et al. [57], contains 102K non-factoid topics with long-form answers. These topics were curated from Bing Search logs and evaluated by GPT-4 on a scale of 0–10 based on eight intrinsic attributes, such as subjectivity and multifacetedness. Notably, unlike TREC-RAGgy 2024, these queries lack relevance judgments. To curate a smaller development subset for a faster evaluation of RAG systems, we employ a sampler designed to maximize diversity based on the eight intrinsic attributes. This is achieved by iteratively selecting the query with the highest l_1 norm in the intrinsic attribute space (of all eight dimensions) relative to the already-sampled set. We refer to the resultant topic set as TREC-Researchy 2024. From Table 2, about 80% of the topics are Knowledge-Intensive, and about 76% are Multi-Faceted, highlighting the need for effective RAG systems. Additionally, 66.5% of topics start with "how" or "why", emphasizing explanatory questions. These distributions suggest that TREC-Researchy 2024 prioritizes complex and multi-dimensional topics.

For the TREC RAG test topics, we released a new and fresh scrape of topics close to the submission period. This approach compiled a fresh and newer set of topics, similar to Rosset et al. [57], thereby minimizing the risk of data leakage and ensuring a fair evaluation with existing commercially available LLMs.

6 TREC RAG Baselines

6.1 Retrieval

Our retrieval module integrates both first-stage retrievers and rerankers. For traditional dual encoders, we employ BM25 and BM25 + Rocchio, available in Anserini [65] and retrieve the top 3000 segments for a given topic. BM25 has proven effective as a first-stage retriever due to its capability to capture lexical overlap, with Rocchio introducing relevance feedback to further refine the search results based on the initial BM25 retrieval.

We further extend our retrieval evaluation by incorporating GTE-L, a dense dual encoder model from Alibaba-NLP [34]. GTE-L is their larger variant comprising 434M parameters, with embeddings of dimensionality 1024, and support for a maximum sequence length of 8196 tokens. This model is particularly suited for long-context retrieval tasks, enabling it to effectively capture relationships across larger text segments. GTE-L is a strong baseline demonstrating state-of-the-art effectiveness in the MTEB [46] benchmark, when it was introduced.

Additionally, we evaluate two other dense dual encoders from Snowflake: ArcticEmbed-M and ArcticEmbed-L [44]. ArcticEmbed-M is a 137M parameter model, generating embeddings with a dimensionality of 768, while ArcticEmbed-L is a larger model with 335M parameters and an embedding dimensionality of 1024. Both models have shown state-of-the-art performance in retrieval tasks, particularly on the BEIR benchmark [61], where they excel in diverse retrieval settings. ArcticEmbed-M and ArcticEmbed-L provide efficient and scalable solutions for semantic retrieval, complementing our first-stage retrieval suite.

For each retrieval method (BM25, BM25 + Rocchio, GTE-L, ArcticEmbed-M, and ArcticEmbed-L), we retrieve the top 3000 segments. To leverage the strengths of multiple retrieval strategies, we employ Reciprocal Rank Fusion (RRF) as a hybrid approach. The RRF technique fuses the retrieval results from BM25 + Rocchio, GTE-L, ArcticEmbed-M, and ArcticEmbed-L, with the aim of producing a better candidate set for reranking, as hybrid approaches have been shown to improve retrieval effectiveness in multiple TREC tracks [32].

Following retrieval, we explore neural reranking models for first-stage reranking. We leverage monoT5-3B, a pointwise reranker with 3B parameters, that demonstrates strong effectiveness in reranking by assigning relevance scores to individual document-query pairs [47,52]. Note that we fuse these results with the results from the prior stage.

Next, we incorporate RankZephyr, a highly effective listwise reranker in Pradeep et al. [54], which takes as input a query and a list of passages and outputs a reordered list based on relevance. RankZephyr is particularly effective in scenarios where contextual relationships between documents are useful for ideal reranking. In our pipeline, we use RankZephyr to rerank the top 100 resultant candidates from monoT5-3B, ensuring a more precise final ranking. Note that we fuse the RankZephyr results with the results from the prior stage.

Additionally, we evaluate RankZephyr$_\rho$, which adopts a progressive reranking strategy. RankZephyr$_\rho$ iteratively refines the ranking of candidate documents over three passes, progressively improving the precision of the final ranked list. This iterative refinement has been shown to enhance retrieval effectiveness, particularly in cases where high-precision ranking is required.

Both monoT5-3B and RankZephyr and also other rerankers such as LRL [41], RankGPT [59] and LiT5 [60], are available through the `rank_llm` package, which we leverage as part of our reranking pipeline in Ragnarök. While budget constraints prevent us from testing every reranking model, support for this functionality continues through Ragnarök. Some of these retrieved results in the case of TREC-RAGgy 2024 can be evaluated after running the MaxP operation to get the representative score for the document. Finally, the top-20 reranked segments are passed on to the next stage, i.e., augmented generation.

Results. Table 3 presents the nDCG@10, MAP@100, and recall@100 for the primary retrieval baselines under consideration. The traditional lexical retrievers BM25 (1a) and BM25+Rocchio (1b) demonstrate comparable scores with only marginal differences. Dense dual encoders (1c, 1d, 1e) exhibit superior effectiveness compared to traditional lexical methods (1a, 1b). Among the dense models, the ArcticEmbed variants (1d, 1e) achieve slightly higher effectiveness metrics than GTE-L (1c). Notably, the application of Reciprocal Rank Fusion across

Table 3. Results on the Document Ranking Task of the TREC-RAGgy 2024.

Model	nDCG@10	MAP@100	Recall@100
Lexical & Dual Encoders			
(1a) BM25	0.4227	0.1561	0.2807
(1b) BM25+Rocchio	0.4188	0.1818	0.3141
(1c) GTE-L	0.5682	0.2162	0.3512
(1d) ArcticEmbed-M	0.5749	0.2349	0.3692
(1e) ArcticEmbed-L	0.5776	0.2277	0.3623
(1f) RRF(1b,1c,1d,1e)	0.6064	0.2592	0.3990
Rerankers			
(2a) RRF(1f, monoT5-3B)	0.6175	0.2708	0.4208
(2b) RRF(2a, RankZephyr)	0.6357	0.2770	0.4208
(2c) RRF(2a, RankZephyr$_\rho$)	0.6317	0.2771	0.4208

these models (1f) yields substantial improvements compared to any individual retriever (1a–e).

The integration of the pointwise reranker monoT5-3B (2a) further enhances retrieval effectiveness across all metrics. Subsequent application of listwise reranking models (2b, 2c) yields additional effectiveness gains. However, the progressive reranking approach does not demonstrate meaningful improvements in high-precision metrics for this particular collection, as evidenced between the multiple and single-pass RankZephyr variants (2c vs. 2b).

6.2 Augmented Generation

Our generation module is designed to support a wide range of both open-source and proprietary models, enabling flexible integration for various use cases. At the core of our open-source support is vLLM [30], an optimized framework for efficient large language model (LLM) inference. vLLM excels in handling resource-intensive models by leveraging tensor parallelism and dynamic batching, ensuring scalability and performance across distributed systems. This makes it well-suited for open-source LLMs, where efficient resource management is critical. In addition, vLLM provides a wide support for text-based LLMs[2] including custom models fine-tuned with LoRA [22].

Our focus is on three popular LLMs: (i) GPT-4o is the latest GPT version from OpenAI [48]; (ii) Command R+ is Cohere's open-source instruction following LLM developed for complex RAG pipelines [13]; (iii) LLaMA-3.1-70B-Instruct is Meta's flagship open-source instruction tuned generative model optimized for multilingual dialogue use cases [17].

Given that Command R+ cites in a span level, we map the citations to their parent sentences. For GPT-4o and LLaMA-3.1-70B, we follow the ChatQA

[2] https://docs.vllm.ai/en/latest/models/supported_models.html.

```
System: This is a chat between a user and an artificial intelligence
assistant. The assistant gives helpful, detailed, and polite answers
to the user's questions based on the context. The assistant should
also indicate when the answer cannot be found in the context.

INSTRUCTION: Please give a complete answer to the question. Cite
each context document that supports your answer within brackets []
using the IEEE format.

QUESTION: {query}

CONTEXTS:
[1] {Passage title}: {Passage text}
[2] {Passage title}: {Passage text}
...
[20] {Passage title}: {Passage text}

INSTRUCTION: Please give a complete answer to the question. Cite
each context document that supports your answer within brackets []
using the IEEE format.
```

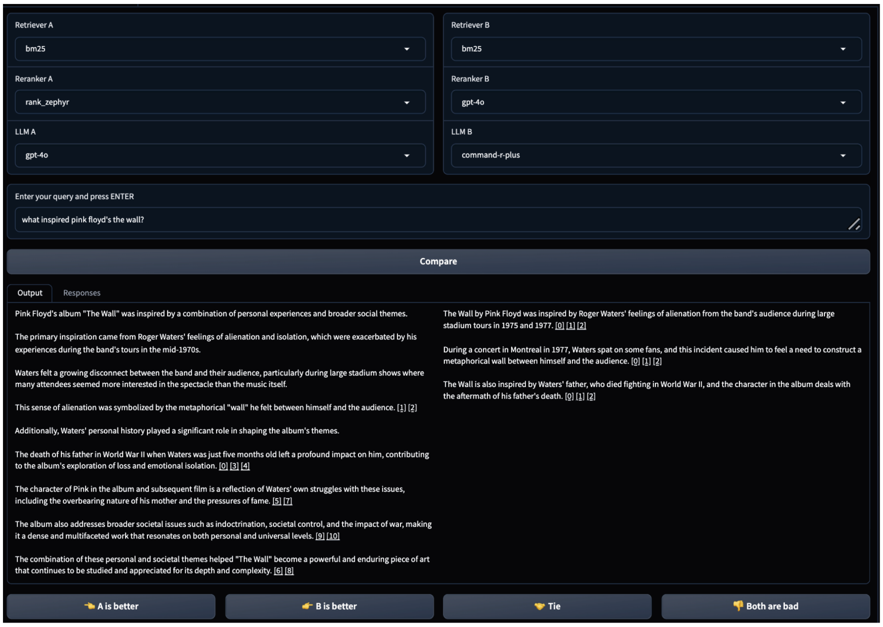
Fig. 2. ChatQA prompt template [40] used for RAG generation with in-text citations with LLaMA3.1-70B and GPT-4o in our Ragnarök framework.

prompt template [40] and cite relevant segments within the text (in-line) using the IEEE format. An example of the prompt template is shown in Fig. 2.

Additionally, Ragnarök incorporates a variety of refined ChatQA prompts specifically tailored for biomedical RAG tasks, particularly within the context of the TREC 2024 Biomedical Generative REtrieval (BioGen) track. These refinements enhance the system's ability to address domain-specific queries effectively in the expected manner and steer the model with expected word counts and sentence structures (each sentence is an independent assertion). Furthermore, the foundational ChatQA template was repurposed for the TREC 2024 Neural Cross-Language Information Retrieval (NeuCLIR) track's report generation task. This adaptability underscores Ragnarök's versatility in prompt management, allowing users to modify prompts with minimal intervention, typically requiring only a single addition within the prompt template file. This streamlined approach ensures that users can efficiently adapt our framework across various domains.

6.3 RAG-Bench Evaluation

Evaluating different RAG answers is challenging as multiple factors within the output response are crucial for effectiveness evaluation. To combat this, recent works rely on an LLM-as-a-judge setup [66], where strong LLM assessors judge the RAG-generated output in a pairwise evaluation style (side-by-side) in a head-on tournament. In our work, we briefly overview our baseline techniques using human evaluators. The Command R+ baseline outputs shorter answers and cites more relevant segments, whereas, the GPT-4o baseline outputs longer and more detailed answers and cites fewer segments. Therefore, for topics in both TREC-RAGgy 2024 and TREC-Researchy 2024, GPT-4o intuitively is the better choice for RAG answer generation. We found that the LLaMA3.1-70B model produces long and detailed answers but sometimes generates (from model parameters) a separate references section, from which it cites. We leave it for future work, to empirically compute the win rates (in %) between our baselines in the RAG-bench evaluation.



Retriever A: bm25
Reranker A: rank_zephyr
LLM A: gpt-4o

Retriever B: bm25
Reranker B: gpt-4o
LLM B: command-r-plus

Enter your query and press ENTER: what inspired pink floyd's the wall?

Pink Floyd's album "The Wall" was inspired by a combination of personal experiences and broader social themes.

The primary inspiration came from Roger Waters' feelings of alienation and isolation, which were exacerbated by his experiences during the band's tours in the mid-1970s.

Waters felt a growing disconnect between the band and their audience, particularly during large stadium shows where many attendees seemed more interested in the spectacle than the music itself.

This sense of alienation was symbolized by the metaphorical "wall" he felt between himself and the audience. [1] [2]

Additionally, Waters' personal history played a significant role in shaping the album's themes.

The death of his father in World War II when Waters was just five months old left a profound impact on him, contributing to the album's exploration of loss and emotional isolation. [0] [3] [4]

The character of Pink in the album and subsequent film is a reflection of Waters' own struggles with these issues, including the overbearing nature of his mother and the pressures of fame. [5] [7]

The album also addresses broader societal issues such as indoctrination, societal control, and the impact of war, making it a dense and multifaceted work that resonates on both personal and universal levels. [9] [10]

The combination of these personal and societal themes helped "The Wall" become a powerful and enduring piece of art that continues to be studied and appreciated for its depth and complexity. [6] [8]

The Wall by Pink Floyd was inspired by Roger Waters' feelings of alienation from the band's audience during large stadium tours in 1975 and 1977. [0] [1] [2]

During a concert in Montreal in 1977, Waters spat on some fans, and this incident caused him to feel a need to construct a metaphorical wall between himself and the audience. [0] [1] [2]

The Wall is also inspired by Waters' father, who died fighting in World War II, and the character in the album deals with the aftermath of his father's death. [0] [1] [2]

Fig. 3. WebUI showcasing the Ragnarök System Arena and the user query, "what inspired pink floyd's the wall?", with answers from two pipelines side-by-side comparing GPT-4o (left) and Command R+ (right).

7 Ragnarök System Arena

Heavily inspired by the success of Chatbot Arena [12,66], a crowdsourcing benchmark WebUI featuring anonymous battles, we extend the concept to multi-stage configurable RAG pipelines with Ragnarök. In the arena, users interact with two unblinded/blinded RAG systems simultaneously, issuing the same topic to both. The participants evaluate and select the pipeline that delivers their most preferred response, with the identities of the modules in the end-to-end pipeline revealed after the voting process in the blinded case. We leverage Gradio [1] to build the WebUI for Ragnarök. Each step of the pipeline uses REST APIs for intercommunication, enabling easy module switching within the pipeline. This modular design simplifies the integration of different retrieval and LLM configurations, enhancing scalability and maintainability.

Figure 3 illustrates an example topic "what inspired pink floyd's the wall?" processed by two different pipelines: Pipeline A, comprising BM25 → RankZephyr → GPT-4o (left), and Pipeline B, comprising BM25 → RankGPT-4o → Command R+ (right) in the unblinded tab. The outputs generated by each pipeline are compared, allowing users to discern which system provided a more satisfactory answer. Note that when the user hovers the mouse over a citation, they can preview the cited segment. We also provide a similar blinded pairwise evaluation interface along with the ability to view the responses in the JSON form (a tab) in the WebUI for Ragnarök.

After reading both outputs, the user has to pick one of the four choices for preference scoring: (i) A is better (ii) B is better (iii) Tie, or (iv) Both are bad. These results from preference scoring are organized into three separate ELO leaderboards, leveraging an SQLite database to track module effectiveness. These leaderboards evaluate language models (LLMs) for augmented generation-only (AG), retrieval models (R), and retrieval-augmented generation pipelines (RAG), providing a comprehensive view of their relative contributions.

8 Ongoing Work

Ragnarök is the first step for the ongoing work in the TREC RAG, by releasing the document collections, development topics, and baseline strategies for participants. We will continue to update the pipelines to include more diverse retrieval models including other sparse dual encoders such as SPLADE-v3 [31] and effective pairwise rerankers [52]. It is also in our critical path to evaluate the retrieved results from submissions to TREC-RAGgy 2024, after pooling, both with NIST annotators and LLM judges [62]. We plan to add additional support for more advanced RAG techniques like SelfRAG [6] and CRAG [64].

The next phase of our efforts will focus on finalizing the evaluation methodology of the generated answer in the RAG output. We are planning to include three evaluation metrics: nugget recall, support, and fluency. Automatic nugget-based evaluation is gaining popularity [2,5,43] and becoming the de facto strategy for RAG evaluation. Building on earlier work by Lin et al. [35], we plan to build a list of nuggets for each question recursively passing through the pooled results. We evaluate each generated answer based on precision and recall metrics. Next, for support evaluation, we plan to measure whether the cited documents sufficiently ground the information present in the answer. Lastly, for fluency evaluation, we measure whether the RAG output is coherent and fluent without any grammatical mistakes.

9 Conclusion

The emergence of retrieval-augmented generation (RAG) has revolutionized modern search systems by allowing real-time data incorporation into large language models (LLMs). Our work presents Ragnarök, a reusable and open-source end-to-end framework designed to provide reproducible baselines and a WebUI serving a RAG battle arena for retriever, reranker, and generation models.

In addition to introducing the MS MARCO V2.1 collection, we carefully curated topics from TREC-DL 2021–2023 and Researchy Questions. The TREC-RAGgy 2024 subset and the corpus can be collectively viewed as an ad-hoc retrieval collection on which we evaluated several state-of-the-art dual encoders and rerankers. Our work also defines I/O specifications to assist users in the RAG paradigm. By tightly integrating with popular retrieval frameworks such as Pyserini and `rank_llm`, Ragnarök ensures seamless usage.

We identify key baselines from industry, including OpenAI's GPT-4o, Cohere's Command R+ and Meta's LLaMA3.1-70B, and provide a qualitative analysis of these baselines on the development topics. Our framework provides a WebUI for head-to-head RAG system comparisons, inspired by recent work such as Chatbot Arena, to facilitate user evaluation and preference scoring.

Finally, by open-sourcing Ragnarök, we aim to standardize RAG applications in preparation for TREC RAG, promoting wider adoption and fostering innovation in the RAG research community. We plan to continuously update Ragnarök to include more advanced retrieval models and RAG techniques, ensuring it remains a valuable tool for researchers and practitioners alike.

Acknowledgments. We thank Ian Soboroff for his support with MS MARCO V2 document collection deduplication, Cohere for providing us with the necessary credits to evaluate Command-R+, and Microsoft for providing Azure credits to evaluate GPT-4o. Additionally, we thank Corby Rosset for the discussions surrounding Researchy Questions [57]. Additional funding is provided by Snowflake.

References

1. Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., Zou, J.Y.: Gradio: hassle-free sharing and testing of ML models in the wild. CoRR arxiv:1906.02569 (2019)
2. Alaofi, M., Arabzadeh, N., Clarke, C.L.A., Sanderson, M.: Generative information retrieval evaluation. CoRR arxiv:2404.08137 (2024)
3. Aliannejadi, M., Abbasiantaeb, Z., Chatterjee, S., Dalton, J., Azzopardi, L.: TREC ikat 2023: the interactive knowledge assistance track overview. CoRR arxiv:2401.01330 (2024)
4. Amouyal, S.J., Rubin, O., Yoran, O., Wolfson, T., Herzig, J., Berant, J.: QAMPARI: an open-domain question answering benchmark for questions with many answers from multiple paragraphs. CoRR arxiv:2205.12665 (2022)
5. Arabzadeh, N., Clarke, C.L.A.: A comparison of methods for evaluating generative IR. CoRR arxiv:2404.04044 (2024)
6. Asai, A., Wu, Z., Wang, Y., Sil, A., Hajishirzi, H.: Self-RAG: learning to retrieve, generate, and critique through self-reflection. CoRR arxiv:2310.11511 (2023)

7. Bajaj, P., et al.: MS MARCO: a human generated machine reading comprehension dataset. CoRR arxiv:1611.09268 (2016)
8. Bolotova, V., Blinov, V., Scholer, F., Croft, W.B., Sanderson, M.: A non-factoid question-answering taxonomy. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022, pp. 1196–1207. ACM (2022)
9. Borgeaud, S., et al.: Improving language models by retrieving from trillions of tokens. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S. (eds.) International Conference on Machine Learning, ICML 2022, Baltimore, Maryland, USA, 17–23 July 2022. Proceedings of Machine Learning Research, vol. 162, pp. 2206–2240. PMLR (2022)
10. Broder, A.Z.: On the resemblance and containment of documents. In: Carpentieri, B., Santis, A.D., Vaccaro, U., Storer, J.A. (eds.) Compression and Complexity of SEQUENCES 1997, Positano, Amalfitan Coast, Salerno, Italy, 11–13 June 1997, Proceedings, pp. 21–29. IEEE (1997)
11. Chase, H.: Langchain (2022)
12. Chiang, W., et al.: Chatbot arena: an open platform for evaluating LLMs by human preference. CoRR arxiv:2403.04132 (2024)
13. Cohere: Introducing Command R+: a scalable llm built for business (2024)
14. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Lin, J.: Overview of the TREC 2021 deep learning track. In: Soboroff, I., Ellis, A. (eds.) Proceedings of the Thirtieth Text REtrieval Conference, TREC 2021, online, 15–19 November 2021. NIST Special Publication, vol. 500-335. National Institute of Standards and Technology (NIST) (2021)
15. Craswell, N., et al.: Overview of the TREC 2022 deep learning track. In: Soboroff, I., Ellis, A. (eds.) Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, 15–19 November 2022. NIST Special Publication, vol. 500-338. National Institute of Standards and Technology (NIST) (2022)
16. Craswell, N., et al.: Overview of the TREC 2023 Deep Learning Track. In: Text REtrieval Conference (TREC). NIST, TREC (2024)
17. Dubey, A., et al.: The Llama 3 herd of models. arXiv:2407.21783 (2024)
18. Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., Auli, M.: ELI5: long form question answering. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019, vol. 1: Long Papers, pp. 3558–3567. Association for Computational Linguistics (2019)
19. Gao, T., Yen, H., Yu, J., Chen, D.: Enabling large language models to generate text with citations. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, 6–10 December 2023, pp. 6465–6488. Association for Computational Linguistics (2023)
20. Gao, Y., et al.: Retrieval-augmented generation for large language models: a survey. CoRR arxiv:2312.10997 (2023)
21. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 3929–3938. PMLR (2020)
22. Hu, E.J., et al.: Lora: low-rank adaptation of large language models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, 25–29 April 2022. OpenReview.net (2022)

23. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. In: Merlo, P., Tiedemann, J., Tsarfaty, R. (eds.) Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021, pp. 874–880. Association for Computational Linguistics (2021)
24. Jimeno-Yepes, A., You, Y., Milczek, J., Laverde, S., Li, R.: Financial report chunking for effective retrieval augmented generation. CoRR arxiv:2402.05131 (2024)
25. Jin, J., Zhu, Y., Yang, X., Zhang, C., Dou, Z.: FlashRAG: a modular toolkit for efficient retrieval-augmented generation research. CoRR arxiv:2405.13576 (2024)
26. Kamalloo, E., Jafari, A., Zhang, X., Thakur, N., Lin, J.: HAGRID: a human-LLM collaborative dataset for generative information-seeking with attribution. CoRR arxiv:2307.16883 (2023)
27. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: nearest neighbor language models. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020. OpenReview.net (2020)
28. Kulkarni, S., Chammas, S., Zhu, W., Sha, F., Ie, E.: Aquamuse: automatically generating datasets for query-based multi-document summarization. CoRR arxiv:2010.12694 (2020)
29. Kwiatkowski, T., et al.: Natural Questions: a benchmark for question answering research. Trans. Assoc. Comput. Linguist. **7**, 452–466 (2019)
30. Kwon, W., et al.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles (2023)
31. Lassance, C., Déjean, H., Formal, T., Clinchant, S.: SPLADE-v3: new baselines for SPLADE. arXiv:2403.06789 (2024)
32. Lassance, C., Pradeep, R., Lin, J.: naverloo @ TREC deep learning and NeuCLIR 2023: as easy as zero, one, two, three—cascading dual encoders, mono, duo, and listo for ad-hoc retrieval. In: Text REtrieval Conference (TREC). NIST (2024)
33. Lewis, P.S.H., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 6–12 December 2020, virtual (2020)
34. Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards general text embeddings with multi-stage contrastive learning. arXiv:2308.03281 (2023)
35. Lin, J., Demner-Fushman, D.: Methods for automatically evaluating answers to complex questions. Inf. Retr. **9**(5), 565–587 (2006)
36. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: a python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 2356–2362 (2021)
37. Lin, S., Hilton, J., Evans, O.: Truthfulqa: measuring how models mimic human falsehoods. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, 22–27 May 2022, vol. 1: Long Papers, pp. 3214–3252. Association for Computational Linguistics (2022)
38. Liu, J.: Llamaindex (2022)

39. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: how language models use long contexts. Trans. Assoc. Comput. Linguist. **12**, 157–173 (2024)
40. Liu, Z., et al.: Chatqa: building GPT-4 level conversational QA models. CoRR arxiv:2401.10225 (2024)
41. Ma, X., Zhang, X., Pradeep, R., Lin, J.: Zero-shot listwise document reranking with a large language model. arXiv:2305.02156 (2023)
42. Malaviya, C., Lee, S., Chen, S., Sieber, E., Yatskar, M., Roth, D.: Expertqa: expert-curated questions and attributed answers. CoRR arxiv:2309.07852 (2023)
43. Mayfield, J., et al.: On the evaluation of machine-generated reports. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (2024)
44. Merrick, L., Xu, D., Nuti, G., Campos, D.: Arctic-embed: scalable, efficient, and accurate text embedding models (2024)
45. Microsoft: Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web (2023)
46. Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: MTEB: massive text embedding benchmark. In: Vlachos, A., Augenstein, I. (eds.) Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, 2–6 May 2023, pp. 2006–2029. Association for Computational Linguistics (2023)
47. Nogueira, R., Jiang, Z., Pradeep, R., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 708–718 (2020)
48. OpenAI: Hello GPT-4o (2024)
49. Overwijk, A., Xiong, C., Callan, J.: Clueweb22: 10 billion web documents with rich information. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022, pp. 3360–3362. ACM (2022)
50. Owoicho, P., Dalton, J., Aliannejadi, M., Azzopardi, L., Trippas, J.R., Vakulenko, S.: TREC CAsT 2022: going beyond user ask and system retrieve with initiative and response generation. In: Soboroff, I., Ellis, A. (eds.) Proceedings of the Thirty-First Text REtrieval Conference, TREC 2022, online, 15–19 November 2022. NIST Special Publication, vol. 500-338. National Institute of Standards and Technology (NIST) (2022)
51. Petroni, F., et al.: KILT: a benchmark for knowledge intensive language tasks. In: Toutanova, K., et al. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021, pp. 2523–2544. Association for Computational Linguistics (2021)
52. Pradeep, R., Nogueira, R., Lin, J.: The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. arXiv:2101.05667 (2021)
53. Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankVicuna: zero-shot listwise document reranking with open-source large language models. CoRR arxiv:2309.15088 (2023)
54. Pradeep, R., Sharifymoghaddam, S., Lin, J.: RankZephyr: effective and robust zero-shot listwise reranking is a breeze! CoRR arxiv:2312.02724 (2023)
55. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)

56. Rosenthal, S., Sil, A., Florian, R., Roukos, S.: CLAPNQ: cohesive long-form answers from passages in natural questions for RAG systems. CoRR arxiv:2404.02103 (2024)
57. Rosset, C., et al.: Researchy questions: a dataset of multi-perspective, decompositional questions for LLM web agents. CoRR arxiv:2402.17896 (2024)
58. Stelmakh, I., Luan, Y., Dhingra, B., Chang, M.: ASQA: factoid questions meet long-form answers. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022, pp. 8273–8288. Association for Computational Linguistics (2022)
59. Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., Ren, Z.: Is ChatGPT good at search? investigating large language models as re-ranking agent. arXiv:2304.09542 (2023)
60. Tamber, M.S., Pradeep, R., Lin, J.: Scaling down, LiTting up: efficient zero-shot listwise reranking with Seq2seq encoder-decoder models. arXiv:2312.16098 (2023)
61. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: a heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Vanschoren, J., Yeung, S. (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual (2021)
62. Upadhyay, S., Pradeep, R., Thakur, N., Craswell, N., Lin, J.: UMBRELA: UMbrela is the (open-source reproduction of the) bing RELevance assessor. arXiv:2406.06519 (2024)
63. Xiong, G., Jin, Q., Lu, Z., Zhang, A.: Benchmarking retrieval-augmented generation for medicine. CoRR arxiv:2402.13178 (2024)
64. Yan, S., Gu, J., Zhu, Y., Ling, Z.: Corrective retrieval augmented generation. CoRR arxiv:2401.15884 (2024)
65. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of lucene for information retrieval research. In: International Conference on Research and Development in Information Retrieval (SIGIR) (2017)
66. Zheng, L., et al.: Judging LLM-as-a-judge with MT-bench and chatbot arena. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, 10–16 December 2023 (2023)