2025 IEEE 22nd International Conference on Software Architecture Companion (ICSA-C) | 979-8-3315-3336-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICSA-C65153.2025.00074

Agentic RAG with Human-in-the-Retrieval

Xiwei Xu*†, Dawen Zhang*, Qing Liu*†, Qinghua Lu*†, and Liming Zhu*†

* CSIRO's Data61, Sydney, Australia
† University of New South Wales (UNSW), Sydney Australia
firstname.secondname@data61.csiro.au

***Abstract*—Retrieval-Augmented Generation (RAG) has emerged as a promising solution to address key challenges faced by GenAI, such as hallucination, outdated or non-removable parametric knowledge, and non-traceable reasoning processes. Existing RAG frameworks introduce dynamism into RAG process through adaptive, recursive and interactive usage of *retriever* and *generator*. More recently, agentic RAG adds another layer of intelligence to RAG by leveraging GenAI agents to further enhance dynamism by autonomously planning the retrieval process as a complex orchestration workflow with various external tools. However, current RAG architectures often overlook the significant role that domain experts can play in the retrieval process, alongside passive knowledge bases. This paper introduces a new paradigm for agentic RAG systems, capable of integrating external passive knowledge bases as well as active domain experts. This integration further enhances the versatility and factual accuracy of RAG systems. The paper discusses the key components of this new paradigm and examines the associated design challenges.**

***Index Terms*—RAG, Agentic RAG, GenAI**

I. Introduction

Generative AI (GenAI) is quickly evolving, enabling the creation of novel content like text, images, audio, and videos by leveraging learned patterns from existing data. However, GenAI models are inherently limited by the timeliness and coverage of the data they were trained on, which means they may lack information from specific domains or more recent developments that are not covered in their training data. Retrieval-Augmented Generation (RAG) has emerged as an effective solution to address these issues. RAG works by retrieving relevant information from external knowledge bases, thereby enhancing the relevance and faithfulness of GenAI outputs. Additionally, RAG allows for continuous updates with the most recent information. The three key components of a RAG system are *retrieval sources*, *retriever*, and *generator* [6].

Figure 1 illustrates the spectrum of dynamism in RAG architectures, showing how it increases with the introduction of more advanced designs and techniques. A basic RAG system involves a static retrieval process, where the retriever and generator are used sequentially. Existing RAG frameworks [20], [6] have evolved to enable more dynamic workflows by incorporating adaptive, recursive, and interactive retrieval processes, including the recursive use of retrievers or generators and multiple rounds of interaction between them. With these advancements in engineering, the unique combination of retrieval sources, retrievers, and generators within a RAG is tailored to complex tasks, enhancing overall performance and quality [20], [6].

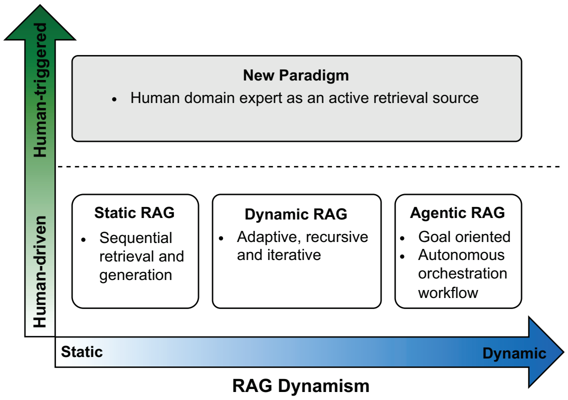
Human-triggered
Human-driven
New Paradigm
Human domain expert as an active retrieval source
Static RAG
Sequential retrieval and generation
Dynamic RAG
Adaptive, recursive and iterative
Agentic RAG
Goal oriented
Autonomous orchestration workflow
Static
Dynamic
RAG Dynamism

Fig. 1. RAG Dynamism Spectrum

More recently, agentic RAG has emerged as a trend in the field, which is an agent-based RAG framework that operates as an orchestrated workflow with potential multiple agents, driven by inferred goals or intent from user inputs. These intelligent agents can tackle complex questions requiring intricate planning, multi-step reasoning, and utilization of tools[1] for web searches, SQL execution, embedding-based retrieval, and graph-based queries. Agentic RAG adds an extra layer of intelligence by incorporating GenAI agents as autonomous decision-makers for knowledge-intensive tasks, further enhancing the dynamism of retrieval processes. These agents autonomously plan the retrieval process as a more complex orchestration workflow, utilizing various tools. However, practical design guidance for agentic RAG remains barely addressed in existing reviews and surveys in the field [6], [7], [18], [20].

Current RAG architectures, whether static or dynamic, often focus solely on passive knowledge bases such as the web, vector databases, knowledge graphs (KG), or tabular data. This approach overlooks the significant role that human domain experts can play in the retrieval process, especially when these

[1] https://www.leewayhertz.com/agentic-rag/#What-is-agentic-RAG

2768-4288/25/$31.00 ©2025 IEEE
DOI 10.1109/ICSA-C65153.2025.00074

experts are the users of RAG systems in advanced professional domains or scientific scenarios.

Through multidisciplinary projects, workshops, and interviews with scientists from fields such as agriculture and biology, we have been working closely to explore and understand their requirements for using GenAI with their domain-specific data sources through RAG to address their daily scientific questions. We found that when domain experts use RAGs, the questions they frame often lack sufficient context. However, even with effective information retrieval, the sources may struggle to cover the latest trends in rapidly evolving areas. Additionally, domain experts may possess *tacit knowledge*—implicit understanding that is difficult to articulate or extract and, therefore, cannot easily be transferred or captured by passive retrieval sources through writing or verbal communication. In a RAG process, we believe that humans with tacit knowledge can be treated as active knowledge sources, enabling the retrieval process to rely on them to extract the contextual information necessary for solving tasks. This paper introduces a new paradigm for agentic RAG incorporating human-in-the-retrieval capabilities. Building on existing GenAI agent reference architectures [17], this paradigm features key components designed to integrate human domain experts as a distinct retrieval source, embedding their tacit knowledge into the existing retrieval sources.

II. New Paradigm for Agentic RAGs

A conceptual architecture of agentic RAG with human-in-the-retrieval is illustrated in Figure 2. It extends the GenAI agent reference architecture [17] with components tailored for RAG scenarios. The agentic RAG and other finer grained agents can use pre-trained or fine-tuned models (as shown in the *GenAI Models* box). When a user's input is received, the *orchestration* module generates a workflow with sub-tasks to answer the user's query. This workflow may incorporate external knowledge if the query cannot be adequately answered by the GenAI model alone. Other modules translate the orchestration module's decisions into specific steps that could be tool callings or actioned by a group of agents, such as retrieval agent, generation agent,and human (indicated by dashed line boxes in Figure 2). The *retrieval* module is equipped with various read-only query tools designed to query different formats of *retrieval sources*. The *generation* module can adopt different strategies to adjust the context retrieved from external sources before passing the information to the GenAI model to answer users. The *guardrail* module includes several mechanisms implemented at different stages of RAG process to ensure the safety and reliability of the retrieval processes. Human-in-the-loop is a common strategy employed in broader GenAI agents [17], where direct integration with humans is used as an intuitive approach to enhancing orchestration capabilities. The *monitorability* module tracks the updates in human feedback and assess their impact during the online RAG process and off-line RAG pipeline.

A. Orchestration

Agentic RAG involves multi-agents to collaboratively determine the optimal moments and context for retrieval, such as adding more steps to the orchestration workflow to autonomously decide when to incorporate external knowledge and when to stop the retrieval and generation process. Consequently, agentic RAG enhances the efficiency and relevance of sourced information. The orchestration workflow calls the GenAI quality monitor function to assess the confidence of the generation process by tracking the probability of the generated terms. The retrieval action is triggered when the probability falls below a certain threshold to gather relevant external information, thereby optimizing the retrieval process. This approach surpasses the static RAG retrieval process by assessing the necessity of retrieval based on varying scenarios [9].

For more complex RAG tasks, the orchestration module empowers the agentic RAG with the ability to break down these tasks into simpler sub-tasks and solve them individually. Existing reasoning strategies on prompt engineering, such as Chain-of-Thought, Tree-of-Thoughts [19] etc., can be integrated into this process. In the context of RAG, Chain-of-Knowledge (CoK) [11] dynamically incorporates grounding information from heterogeneous retrieval sources using the basic CoT strategy to iterative refine rationales step by step, adapting the reasoning process to the domain identified initially based on the question from the user. Similarly, another Chain-of-Knowledge (CoK) [16] extends the CoT reasoning workflow by utilizing a set of exemplars that integrate external structured knowledge evidence. This CoK embeds a list of evidence triples that reflect the overall reasoning evidence from the query towards the answer, as well as explanation hints that provide insights into the reasoning process.

B. Retrieval

The retrieval component is mainly implemented based on a series of query tools that retrieve information from external sources in different data structures and formats without altering their states. The retrieval function can be also implemented as an agent that decides the best tools to be called and generates queries that are specifically tailored to each retrieval source. Semantic similarity search is used to query vector databases. It compares the similarity between the query vector and the vectors of chunks within the vector database, retrieving the top chunks with the highest similarity to the user query. SQL queries are employed to retrieve data from tabular databases, while SPARQL queries are used to access knowledge graphs (KGs). Web search tools can also call RESTful APIs online. The query tools are extendable to support other data structures, such as Cypher for graph structures stored in Neo4j[2]. Additionally, the basic query tools have been encapsulated and integrated into other augmentation mechanisms for both retrieval and generation, for example, GraphRAG [5] is based on a graph data structure and provides graph query tools.

[2] https://neo4j.com/

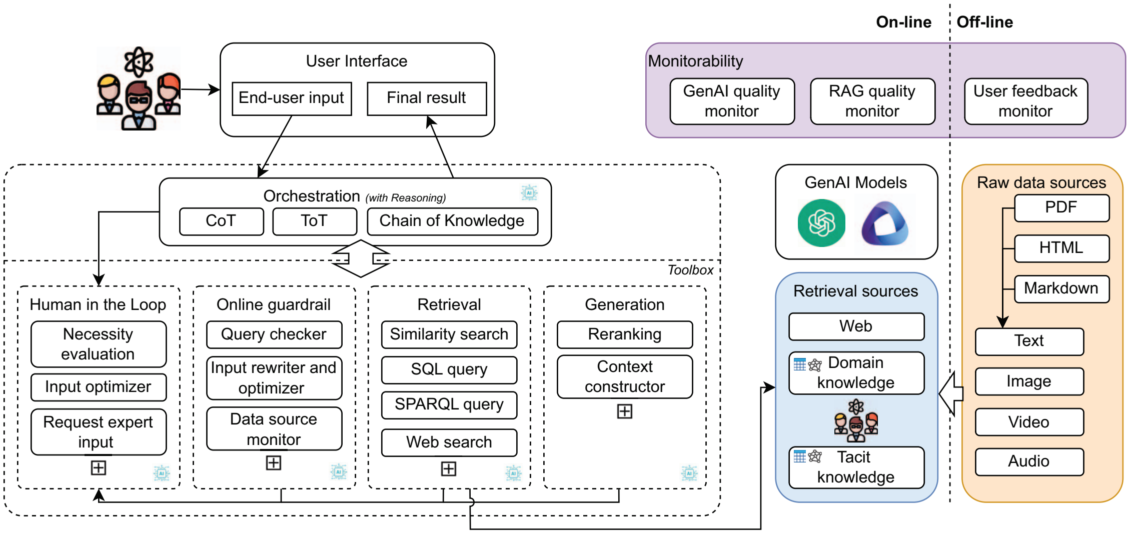

Fig. 2. Conceptual Architecture for Agentic RAG with Human-in-the-Retrieval

C. Generation

The generation module then adjusts the context retrieved from external sources before sending it to the GenAI model to answer users. Similarly as above, generation function could be implemented as a generation agent that calls the tools. Redundant information can negatively affect the final result generation, and excessively long contexts can introduce more noise [12], reducing the generator's ability to focus on key information. The context constructor is designed to filter and structure context information to optimize generation. There are multiple tools that can be actioned by the generation agent. For example, the reranking tool reorders retrieved information to prioritize the most relevant context first [20]. This effectively narrows down the pool of retrieved information and delivers refined information for more precise processing [21]. Existing re-ranking methods, commonly used in recommender systems, might be applicable to RAG scenarios [13].

D. Guardrail

Guardrails [4] are essential to ensure safe and responsible behavior. Guardrail (agent) specifically monitors and controls the tool interactions of agentic RAG, ensuring operations remain within ethical and legal boundaries. As agentic RAG autonomously retrieves external data and knowledge, guardrails are primarily implemented to mitigate injection attacks. For example, when tabular data is retrieved, any SQL query generated by a GenAI model must be rewritten into a semantically equivalent query that operates only on data the user is authorized to access [15]. Alternatively, a SQL query checker can be called to intercept and filter the generated SQL query before it is submitted to the database. In certain cases, the SQL queries are enforced to have read-only permissions when the query execution is not intended to modify the data source. Additionally, a data source monitor can be used to track the sources and quality of the data being retrieved and integrated into the system. The agent reviews the execution of queries; if any suspicious content is detected, the query execution is aborted.

E. Retrieval Sources

Domain knowledge in RAG can be stored in various data structures, with embeddings being one of the primary forms, typically stored in vector databases. Knowledge graphs (KGs) and tabular data have also become essential data structures for RAG. Knowledge graphs are particularly effective in representing the relationships of data points within a broader context, as well as in relation to other data points. KGs are structured as triples, offering a well-organized representation of information that enhances the retrieval process. This structured format allows for the explicit definition and capture of relationships between entities, providing improved fidelity and interpretability. As a result, KGs are especially suitable for applications that require complex inference and reasoning. Existing knowledge graphs, such as Wikidata[3], can be accessed through RESTful APIs for querying.

The web is a crucial data source [8]. Utilizing web search tools like the Google or Bing APIs allows models to access a vast, up-to-date repository of information, which is invaluable for retrieving relevant data for each query. WebGPT [14] extends the capability of GPT-3 with external search engine during text generation. Special tokens are used to facilitate actions in the orchestration workflow, such as search engine queries, browsing results, and quoting sources.

[3] https://www.wikidata.org/

As discussed in Section II-F, in addition to these passive retrieval sources, humans — particularly scientists or professionals in advanced fields — also serve as active retrieval sources that are able to provide just-in-time feedback across the agentic RAG process.

Through a off-line pipeline, raw data can be collected and extracted in various formats, including text, images, audio, and videos. This diverse data is then used to construct vector databases and KGs, both of which are capable of supporting multimodal data. Text data comes in different formats, such as PDFs, HTML pages, Word documents, and Markdown files, which can be converted into a unified text format for further processing and analysis. This conversion process ensures that the data is efficiently indexed, searched, and utilized by agentic RAG.

F. Human-in-the-retrieval

Existing agentic RAGs [1] have not yet considered incorporating human feedback as a potential retrieval source, in addition to the passive retrieval sources discussed in Section II-E. Human feedback, as a subjective signal, can effectively help GenAI align with human values and preferences. When GenAI agents are empowered to actively request feedback from humans, the agent can incorporate human feedback into its prompts, providing just-in-time feedback and leading to more informed planning and reasoning.

In the context of RAG, particularly in scientific scenarios or advanced professional domains such as law, there may be cutting-edge technologies, innovative knowledge or tacit knowledge that are not documented or readily accessible and exist solely in the minds of experts or specialists. For instance, when scientists dealing with tabular data, the schema may be provided, offering a structural overview. However, the implicit reasoning behind why the data was organized in a specific structure, including the nuances of the semantic meanings between columns, may remain unclear. Characteristics of certain columns, such as the uniqueness of values, anomalies, or other implicit attributes, might not be immediately evident. Additionally, contextual information, like the exponential relationships within the data, might only become apparent during the analysis process, rather than being provided upfront. Furthermore, there is specialized knowledge required for effectively analyzing the data that only experienced scientist possess. This expertise includes intuitive understanding of data patterns, the significance of specific variables in relation to the carefully-crafted experiments, and the interactions between them. These underlying logic and contextual insights are often crucial for accurate interpretation, and without them being explicitly available, it can obscure the effective analysis of the data.

In the domain of law and regulation, similar challenges arise where tacit knowledge plays a key role in decision-making. For example, in legal systems, the appellate process allows a losing party to seek a discretionary review from a higher court. Similarly, AI-based decisions could also be subject to appeal [2]. This requires human involvement in the review process to offer insights into unique circumstances that pattern-based AI systems may overlook. These insights should be documented and leveraged to improve future QA and decision making. For example, the EU's Digital Services Act [3](Article 21) mandates that online platforms establish out-of-court dispute resolution mechanisms to address disagreements between users and platforms. In this context, tacit knowledge specific to individual cases serves as a crucial retrieval resource for preventing future disputes from the platforms.

In the domain of design, LLMs are increasingly used by designers to leverage design-relevant image data through RAG, focusing on specific products, industries, or manufacturer. Tacit knowledge in this context reflects the designer's unique attributes, preferences and the signature design styles. For example, interactions between designers and LLMs can be personalized by synthesizing design prompts in various sequence and patterns. Personalization involves embedding the designer's reasoning into the LLM's processes, including abductive, deductive, inductive, analogy-based, constraint-based, case-based, and visual reasoning approaches. By incorporating these established design reasoning patterns and frameworks, LLMs can better emulate the cognitive processes of designers. This personalized reasoning patterns is further refined using image data retrieved from the external design-relevant sources.

In such cases, when experts or scientists use agentic RAG, they can become active retrieval sources, accessed through an agent with expert input calling. Incorporating human-in-the-loop in the online RAG process necessitates redesigning and extending both the online and off-line RAG pipelines to effectively integrate valuable but fragmented human feedback.

G. Monitorability

Monitorability is a crucial quality requirement in ML software systems [10]. In the context of human-in-the-retrival, it is essential to consistently monitor human feedback and assess its impact on the ongoing RAG process. These evaluations are also vital for ensuring the RAG process maintains high overall quality, especially when tacit knowledge from human experts has not yet been captured by existing passive retrieval sources. Tracking updates in human feedback and assessing their impact on the online RAG process introduce added complexity to both the online and off-line RAG pipelines.

III. Design Challenges

The main challenge for human-in-the-retrieval is to autonomously determine when human input is necessary. Whenever AI is used to analyze the intermediate result, humans can also fulfill that role. The key is to involve humans in the most appropriate situations, such as handling critical information, boundary cases, very close rankings, or high uncertainty in an AI step. This decision-making process adds an additional layer of necessity evaluation to the whole RAG process, effectively incorporating human expertise as a retrieval source.

As discussed in Section II-D, a challenge lies in the potential need for an input optimization tool to refine human's feedback/input, making it a valuable intermediate input for the ongoing retrieval process.

Another challenge is integrating human-in-the-loop into the orchestration, generation, and online guardrail modules, particularly in scenarios with high uncertainties. This requires expert knowledge to assess workflow quality, evaluate retrieved context, and ensure the correctness of query formulation. Human adjustments following these evaluations are crucial for improving the overall quality of the retrieval processes.

There are two primary design decisions for storing human feedback as a passive retrieval source for future use. The first is whether to integrate fragmented tacit knowledge into the existing domain knowledge or store it separately. Such intermediate knowledge contributed by human could be integrated into relevant existing knowledge sources, such as providing semantic meaning for specific data columns in a tabular structure or adding missing relationships in a knowledge graph. Alternatively, if the feedback contains confidential or sensitive information, such as unpublished ideas in scientific scenarios, it might be stored separately as a distinct retrieval source.

The second design decision is the format or data structure of the human feedback, particularly if it is stored separately. Deciding whether to embed the feedback in a vector database or organize it within a knowledge graph depends on the nature and volume of the feedback. Embedding might be suitable when the feedback is limited and isolated, but as it grows, constructing a data structure that captures relationships between different pieces of feedback could become necessary.

Design decisions regarding monitorability in the off-line data pipeline involve identifying the components responsible for assessing the quality of the on-line RAG process and clearly defined quality metrics, such as relevance, faithfulness, and factuality.

IV. Conclusion

Despite their potential, current agentic RAG often overlooks the important role that domain experts can play in the retrieval process to inject tacit knowledge into the reasoning process, alongside passive knowledge bases. This paper introduces a reference architecture for an agentic RAG system that includes both passive knowledge bases and active domain experts, further enhancing the versatility and accuracy of the retrieval process. The integration of human-in-the-loop offers practical design insights on how to involve end users as active retrieval sources within the RAG process. It also highlights the challenges of implementing an online human-in-the-loop RAG process and updating retrieval sources off-line to incorporate the tacit knowledge provided by end users.

References

[1] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[2] I. G. Cohen, B. Babic, S. Gerke, Q. Xia, T. Evgeniou, and K. Wertenbroch. How ai can learn from the law: putting humans in the loop only on appeal. *npj Digital Medicine*, 6(1):160, 2023.

[3] E. Council. Regulation (eu) 2022/2065 of the european parliament and of the council of 19 october 2022 on a single market for digital services and amending direcfive 2000/31/ec (digital services act), 2022.

[4] Y. Dong, R. Mu, G. Jin, Y. Qi, J. Hu, X. Zhao, J. Meng, W. Ruan, and X. Huang. Building guardrails for large language models, 2024.

[5] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson. From local to global: A graph rag approach to query-focused summarization, 2024.

[6] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, 2024.

[7] Y. Hu and Y. Lu. RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing, 2024.

[8] C. Huyen. Building A Generative AI Platform. https://huyenchip.com/2024/07/25/genai-platform.html, 2024. [Online; accessed 01-Augest-2024].

[9] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig. Active retrieval augmented generation, 2023.

[10] G. A. Lewis, I. Ozkaya, and X. Xu. Software architecture challenges for ml systems. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 634–638. IEEE, 2021.

[11] X. Li, R. Zhao, Y. K. Chia, B. Ding, S. Joty, S. Poria, and L. Bing. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources, 2024.

[12] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the Middle: How Language Models Use Long Contexts, 2023.

[13] X. Liu, G. Wang, and M. Zakirul Alam Bhuiyan. Re-ranking with multiple objective optimization in recommender system. *Transactions on Emerging Telecommunications Technologies*, 33(1):e4398, 2022.

[14] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.

[15] R. Pedro, D. Castro, P. Carreira, and N. Santos. From Prompt Injections to SQL Injection Attacks: How Protected is Your LLM-Integrated Web Application?, 2023.

[16] J. Wang, Q. Sun, X. Li, and M. Gao. Boosting language models reasoning with chain-of-knowledge prompting, 2024.

[17] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen. A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science*, 18(6), Mar. 2024.

[18] X. Wang, Z. Wang, X. Gao, F. Zhang, Y. Wu, Z. Xu, T. Shi, Z. Wang, S. Li, Q. Qian, R. Yin, C. Lv, X. Zheng, and X. Huang. Searching for Best Practices in Retrieval-Augmented Generation, 2024.

[19] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

[20] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui. Retrieval-augmented generation for ai-generated content: A survey, 2024.

[21] S. Zhuang, B. Liu, B. Koopman, and G. Zuccon. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking, 2023.