Towards Reliable Agents: Benchmarking Customized LLM-Based Retrieval-Augmented Generation Frameworks with Deployment Validation

Kevin Wang
University of British Columbia
kevinsk.wang@ubc.ca

Karel Harjono
University of British Columbia
harjono@student.ubc.ca

Ramon Lawrence
University of British Columbia
ramon.lawrence@ubc.ca

Abstract

The emergence of Large Language Models has created new opportunities for building agent applications across various domains. To address the lack of targeted open benchmarks for agent frameworks, we designed a benchmark that features domain-specific, small knowledge bases, and includes a diverse set of questions categorized by type, such as simple, multi-hop, aggregation, and reasoning questions. We evaluated OpenAI's Assistants API versus a RAG assistant built with Langchain and deployed a RAG system based on benchmark insights as a course assistant over a two-year span in a computer science course. Our findings reveal how domain-specific retrieval impacts response accuracy and highlight key challenges in real-world deployment. Notably, in smaller agentic systems with constrained knowledge bases, the primary challenge shifts from retrieval accuracy to *data availability* in the knowledge bases. We present insights from both benchmark evaluation and real-world usage data to guide the development of more reliable and effective agentic applications.

1 Introduction

Intelligent agents and customized assistants are becoming increasingly vital across diverse domains, fundamentally changing how organizations interact with information and users. These agents understand their environment and leverage available tools. The applications span numerous sectors: customer support agents handling product inquiries, educational tutors providing personalized learning guidance, healthcare assistants supporting medical documentation, legal assistants analyzing case documents, and financial advisors processing market reports. These domain-specific agents offer end users more accurate, grounded, and tailored solutions compared to generic language models. To help build these applications, companies from big providers like OpenAI's Assistants API and IBM's WatsonX to frameworks like Langchain all provide services to build agents, combining retrieval/file search, web search, code interpreters, and other tools to build 'all-aware' agents. For many use cases, retrieving relevant information is critical.

Despite the growing popularity of agents, there is a lack of benchmarks specifically tailored to evaluate frameworks for adopters to compare commercial and custom systems. Existing benchmarks for general-purpose RAG systems, such as CRAG (Yang et al., 2024), RGB (Chen et al., 2024), MultiHop-RAG (Tang and Yang, 2024), and CRUD-RAG (Lyu et al., 2024), often rely on large-scale, dynamically changing knowledge bases like search APIs and news articles, limiting reproducibility. Assistant RAG systems typically query a much smaller knowledge base, which introduces distinct challenges in ensuring domain expertise and alignment with the content. A benchmark for these systems should evaluate how effectively they utilize the available documents to enhance their responses and maintain alignment with the provided content.

In this paper, we address this gap by devising a comprehensive end-to-end benchmark that features domain-specific, small knowledge bases, and includes a diverse set of questions on the knowledge bases categorized by type, such as simple, multi-hop, aggregation, and reasoning. We evaluated the benchmark using OpenAI's Assistants API and a RAG assistant built with Langchain.

We deployed the assistant RAG system for course support in the form of an information retrieval chatbot to investigate practical challenges and considerations in deploying such applications. The user interface allows questions to be posed in a conversational way, and the LLM is used to summarize top search results and display them in an integrated fashion for users. This deployment allows observing user interactions, gathering insights and creating recommendations for best practices.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 655–661

This work answers the research questions:

1. **Comparative RAG Benefits:** Which domains and use cases benefit most from RAG implementation, and when is the additional complexity justified by improved performance?

2. **Real-world Performance:** How does a RAG pipeline perform in a real-world setting as a student service chatbot with end users?

3. **Implications based on benchmark and real-world performance:** How can we improve the pipeline to address common challenges in assistant RAG systems?

The paper contributes by the introduction of a benchmark for evaluating frameworks to build customized RAG systems and identifying optimization challenges for real world applications through a two year evaluation of a deployed RAG system built with Langchain.

2 Background

2.1 RAG-based Assistants

There are many retrieval based assistants in customer service (Pandya and Holia, 2023), which integrate information retrieval with large language models to design chatbots for customized help. Some optimization methods for LLM-based RAG systems in specific domains (Zhao et al., 2024) include optimizing the number of documents retrieved and how they influence generation. These frameworks have been deployed and evaluated in many educational contexts for customized assistants for specific courses where course documents are stored in a knowledge base (Wang et al., 2023; Neupane et al., 2024; Goel and Polepeddi, 2018). Other agents leverage different formats of knowledge bases, such as REPOFORMER, an adaptive retrieval strategy for repository-level code completion (Wu et al., 2024).

2.2 RAG

2.2.1 General-purpose RAG

RAG was designed initially to augment LLMs in the context of seq2seq models such as BART (Lewis et al., 2020), where large knowledge bases such as Wikipedia is used before queries are sent to BART as vectors. However, focus has been shifted to RAG as a general idea where a database is used in conjunction with an LLM, which will receive retrieved relevant information from the database together with the original prompt.

2.3 RAG Evaluation

Chen et al. (Chen et al., 2024) devised RGB, a RAG specific benchmark to evaluate LLMs' ability to handle context that can include noise, counterfactual content, and negative rejection. The tests are generated from prompting ChatGPT together with related news articles. They asked ChatGPT to generate test cases and checked the test cases manually. During tests, Google Search API is used to retrieve relevant information to accompany the queries. Similarly, RECALL was introduced to focus on RAG systems efficacy when dealing with counterfactual knowledge in context. Results show that LLMs are easily influenced by counterfactual information (Liu et al., 2023). CRAG, produced by Meta, creates custom test sets. Instead of focusing on a LLM's ability to parse context, CRAG aims to test on 3 areas: web retrieval summarization, knowledge graph aided retrieval and web retrieval augmentation, and end-to-end RAG. The retrieval component uses the brave search API (Yang et al., 2024).

A recent benchmark, DomainRAG, leverages domain specific context instead of large databases like Wikipedia. However, they set up test cases with preset documents, which does not evaluate the retriever component (Wang et al., 2024).

2.3.1 Evaluating Assistant RAG Systems

Evaluation of assistant RAG systems is focused on providing frameworks, metrics, and methods. IBM released InspectorRAGet and Meta produced Comprehensive RAG Benchmark systems. InspectorRAGet, like RAGAS (Es et al., 2023), aims to provide a platform for which metrics of evaluation and a pipeline is provided. Langchain provide their own platform, LangSmith, that evaluates assistant RAG systems by customizing test cases[1].

3 Methodology

3.1 Quantitative Evaluation of Pipeline

Our pipeline for producing the benchmark data is in Figure 1 including LLM generation of test cases, auto-evaluation, and one round of human checking.

[1] https://www.langchain.com/langsmith

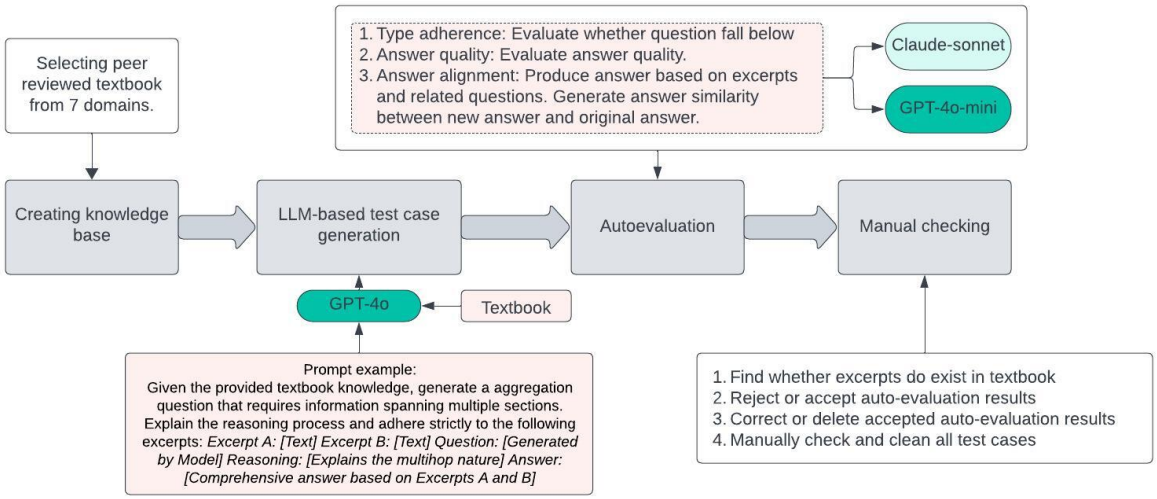

Figure 1: Pipeline for benchmark construction

3.2 Creating Test Cases

We collected 7 textbooks of different domains spanning different levels in higher education. These textbooks are: Business Law I, Calculus III, Microbiology II, Computer Networks: A Systematic Approach, Introduction to Philosophy, Psychology II, and World History II: From 1400. Computer Networks is written by Larry Peterson and Bruce Davie. The rest of the textbooks are from OpenStax. All textbooks used are under CC BY 4.0.

3.2.1 Test case generation

The test cases are generated to have questions and answers closely adhering to the knowledge base. We prompt OpenAI's GPT-4o to generate test cases, using experience from previous work (Chen et al., 2024; Liu et al., 2023; Wang et al., 2024; Friel et al., 2024). The test cases are generated using GPT-4o to closely adhere to the knowledge base. We categorized questions into six types: simple (single-concept questions), aggregation (requiring synthesis of information across multiple sections, such as comparing different antibody types), computation (mathematical operations), reasoning (requiring logical deduction and analysis of implications, like evaluating impacts of cultural awareness), false premise, and multi-hop questions. This categorization helps evaluate different aspects of RAG system performance in real-world scenarios. The benchmark[2] and related code is open-source.

In Figure 1, the outline of the prompt for generating multihop questions is shown. Having the LLM include the reason for why the question falls into the specific question type increases accuracy, and the excerpts allow humans to fact check the questions and ensure question quality.

3.2.2 Auto-Evaluation

We evaluated the benchmark using a baseline PGVector implementation with the LangChain library and OpenAI embeddings. The system performs recursive text splitting with 1000-character chunks and a 20-character overlap, leveraging both ChatGPT and locally hosted LLMs on an Nvidia RTX 6000 GPU. Our evaluation framework employs three key metrics to compare generated responses against ground truth answers:

- **TF-IDF:** Measures lexical similarity by computing cosine similarity between the ground truth and generated responses based on term frequency-inverse document frequency (TF-IDF) representations.
- **Similarity:** Computes cosine similarity between the embeddings of ground truth and generated responses using OpenAI's text-embedding-ada-002 model.[3] Compared to TF-IDF, this metric captures semantic relationships beyond surface-level word overlap.
- **Correctness:** Assessed using `Ragas` RAG evaluation's factual correctness metric (Es et al., 2023) and using the GPT-4o-mini model

[2]The benchmark and code are available at https://github.com/wskksw/agentic_system_bench.git

[3]Embedding introduced at https://openai.com/index/new-and-improved-embedding-model

as an LLM-based judge, following the protocol in (Zheng et al., 2024). Each factual statement in the AI-generated response is categorized as True Positive (TP), False Positive (FP), or False Negative (FN) relative to the ground truth. The correctness score reflects overall alignment with the reference answer.

3.3 Deployed System

We designed an interface that was hosted on a student support platform (Wang and Lawrence, 2024) and deployed in a computer science course at the University of British Columbia. The RAG pipeline for customization follows the experimental design shown in ChatEd (Wang et al., 2023). To enable effective retrieval in conversations, a summarizer prompt is used to rephrase conversations, which is used to similarity search for relevant chunks.

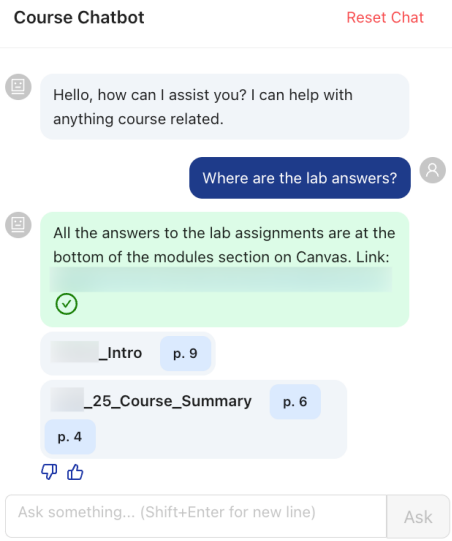

Figure 2: Second version user interface

Interaction results were collected from actual student interactions with the assistant. The first deployed version had a basic chatbot interface, while the second version provided a customizable interface for verifying, suggesting, and editing answers. The system included a similar question feature where questions that had high similarity with previous questions reused answers instead of going through the pipeline. During the first iteration, ChatGPT 4 was used, while ChatGPT 4o-mini was used in the second iteration. Results are evaluated in different metrics by course teaching assistants.

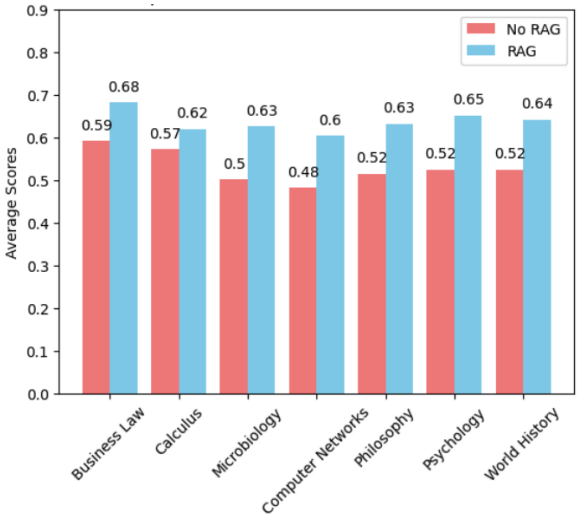

Figure 3: Comparison of LLM and RAG assistants across domains

4 Results

4.1 Comparison of RAG Systems

We are interested in how our benchmark can evaluate different assistant RAG systems. Table 1 shows the performance improvement of using RAG compared to using the LLM only. When comparing the Assistants API from OpenAI to the baseline assistant RAG system, the baseline RAG system performed better, especially in TF-IDF as seen in Table 2. Both RAG systems have an performance increase compared to the same model without RAG.

Auto-evaluation (step 3) for answer alignment also provides another important insight: how well can an LLM perform with 'gold' context. Claude 3.5 sonnet's average answer similarity score is **0.913**, and GPT-4o-mini at **0.886**, both of which are much higher than scores of end-to-end results shown in Table 2. This suggests high potential of optimization of assistant RAG systems to retrieve better context in a specialized knowledge base.

4.1.1 Performance across Domains

Figure 3 demonstrates that the baseline RAG system enhances the performance of LLMs on the benchmark across various fields. The figure shows the average performance in each domain over all LLMs tested (gemma2, llama3.1, GPT-4o). The improvement in Calculus was the least significant. This is likely because Calculus questions, such as "How do you find the distance from a point to a plane?" tend to have straightforward answers that are consistent across different textbooks and online resources. In contrast, questions from fields like

	Assistant RAG Systems			LLM		
Model	**TF-IDF**	**Similarity**	**Correctness**	**TF-IDF**	**Similarity**	**Correctness**
gemma2:27b	0.487	0.847	**0.578**	0.375	0.811	0.534
gemma2:9b	0.490	0.847	0.565	0.364	0.804	0.514
llama3.1:70b	0.516	0.835	0.547	0.423	0.822	0.505
llama3.1:8b	0.513	0.836	0.518	0.432	0.814	0.453
GPT-4o	**0.547**	0.851	0.542	**0.464**	0.846	**0.543**
GPT-4o-mini	0.535	**0.854**	0.556	0.460	**0.856**	0.523

Table 1: Comparison of Non-RAG and RAG Systems with our implementation

RAG System	TF-IDF	Similarity	Correctness
Assistants API (By OpenAI)	0.483	0.851	**0.557**
Baseline RAG	**0.535**	**0.854**	0.556

Table 2: Comparison of RAG Systems with Model GPT-4o-mini

Business Law, such as "What is the ultimate goal of the American legal system?" show more variation. For this question, the textbook specifies that the goal is the "common good", while GPT-4o without any contextual information states that it is "justice". This highlights how assistant RAG systems can be more beneficial in domains where the answers are less standardized and more context-dependent.

4.1.2 Alignment

Assistant RAG systems are shown to be more aligned with ground truth across different models, and enhance local models over OpenAI models. That aligns with expectations, as local models have less parameters and knowledge than OpenAI, and thus might benefit more from extra context.

For the example test case question "What are some types of evidence used in philosophical arguments, and how do they contribute to the strength of these arguments?", the ground truth is compared to systems that all used GPT4-o-mini in Table 3. The baseline RAG system's answer is significantly closer to the ground truth. We highlighted points in the ground truth answer that are in the generated answers. In this case, Assistants API does not perform as well as the baseline RAG, but better than the LLM-only. The observation is backed up by metrics. For the LLM-only answer, the average of the three metrics (TF-IDF, similarity, correctness) is 0.504, whereas the same score is 0.734 for the baseline assistant RAG system and 0.618 for Assistants API.

The baseline assistant RAG system is able to retrieve useful sources for answering the question. This test case shows that an assistant RAG system can potentially increase the alignment of answers with uploaded documents by a significant amount. Interestingly, the RAG-enhanced answer still includes logic in place of intuition from the textbook. We presume that is because of noise in the context.

Assistants API does not directly return cited chunks of information or open source their pipeline, so we do not have information on specific information it retrieved from the file search.

4.1.3 Performance across Question Types

In Figure 4, we observe that false premise questions perform the worst overall, which is consistent with previous findings (Yang et al., 2024). Simple questions improved the most as expected. Simple questions that focus on one specific concept are much more likely to retrieve the 'gold' context from the documents, whereas other types of questions such as the multi-hop example would benefit from a more complex process.

4.2 Real-world Performance

To evaluate the efficacy of our pipeline in a real-world setting, we deployed the system as a student service chatbot interfacing with end users. The deployment was conducted in two phases: an initial version in 2023 and an improved version in 2024. This section presents a comparative analysis of these deployments, highlighting key performance metrics, methodological adjustments, and qualitative observations.

In the first deployment phase in 2023, the chatbot handled a total of **75 queries**. For the subsequent deployment in 2024, there were **451 queries**.

We assessed Question-Answer (QA) interactions

Ground Truth	Common sense, Experimental results, Findings from other disciplines, Experimental philosophy, and Historical insights
LLM-only	Logical Reasoning, Thought Experiments, **Historical Examples**, Intuition and **Common Sense**, Empirical Evidence, Counterexamples, and Expert Testimony
Baseline RAG	**Common Sense**, **Experimental Philosophy**, **Results from Other Disciplines**, Logic, and **History**
Assistants API	**Common Sense**, **Experimental Philosophy**, **Results from Other Disciplines**, Logic, and Intuition

Table 3: Alignment of answers on philosophy question

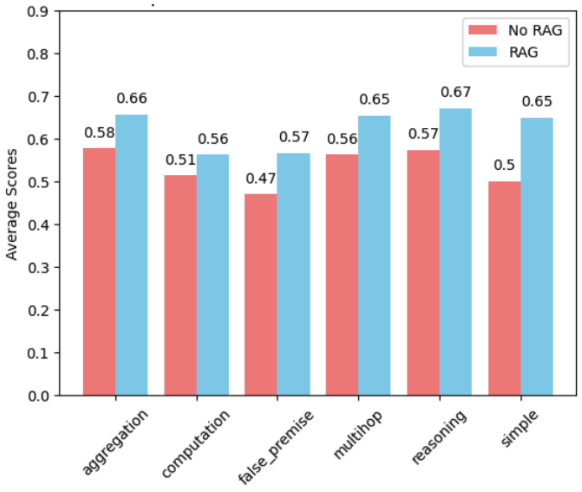

Figure 4: Comparison of LLM and RAG assistants across question types

using four key metrics for real-world deployment effectiveness. From a teaching assistant's perspective, we evaluated whether responses were helpful in resolving user queries. We identified questions requiring additional knowledge base context for accurate responses, flagged potentially harmful queries that could elicit misleading answers, and classified invalid questions that are not answerable.

Quantitative Results Table 4 summarizes the performance metrics for both deployment versions.

Table 4: Chatbot Performance Metrics

Metric	2023 (n=75)	2024 (n=451)
Helpful Answers*	53.2%	66.9%
Needing Context	72.2%	86.3%
Harmful/Wrong	10.1%	6.2%
Invalid	21.5%	13.5%

*Excluding Invalid Questions

Improvements from 2023 to 2024 The 2024 deployment exhibited significant improvements through two key adjustments. First, enhanced prompt engineering introduced specific instructions to prevent pseudo-helpful answers and implemented separate strategies based on question types. Second, a question repository implementation was introduced to handle repetitive queries, utilizing cosine similarity (95% threshold) with 1536-dimensional vector representations, resulting in 20.84% of questions being automatically addressed from previous responses.

Several qualitative insights emerged from the deployments. The chatbot encountered a wide range of query types, from factual inquiries to debugging assistance and system-related questions. This diversity underscores the need to integrate more agentic patterns to enhance the pipeline. Additionally, a significant portion of questions lacked sufficient context, emphasizing the importance of expanding the knowledge base through iterations. Lastly, while harmful responses decreased from 10.13% to 6.21% in the second iteration, their potential impact remains a critical concern for this use case and many other applications.

4.3 Implications based on Benchmark and Real-world Performance

Our benchmark analysis reveals several key insights about RAG systems. First, RAG significantly enhances LLM performance while serving as an effective tool for localized alignment. The effectiveness of RAG varies notably across domains and question types, with simpler, fact-based queries showing the most improvement.

A critical finding is that traditional retrieval optimization techniques, such as reranking, provide minimal benefits when working with specialized, small knowledge bases. Instead, the primary performance bottleneck is the availability of relevant context for most queries. This is evidenced by our comparison between OpenAI's Assistants API (employs more advanced retrieval techniques) and the baseline RAG system - while showing simi-

lar performance with available gold context in our benchmark, real-world deployment revealed that insufficient relevant context often results in plausible but potentially misleading responses.

The gap between benchmark performance (where gold context exists) and real-world performance suggests two key areas for improvement: (1) expanding knowledge base coverage for domain-specific applications, and (2) developing better mechanisms to identify when retrieved context is insufficient for generating reliable responses.

5 Conclusions and Future Work

We introduced an open benchmark for evaluation of agentic behavior in frameworks for customizing LLMs. Our iterative deployments revealed several crucial areas for future development: implementing escalation mechanisms for unresolved queries, developing pipelines for dynamic database expansion based on query patterns, and enhancing agentic solutions through improved tool integration and adaptive retrieval strategies.

References

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking Large Language Models in Retrieval-Augmented Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated Evaluation of Retrieval Augmented Generation. *arXiv preprint arXiv:2309.15217*.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems. *arXiv preprint arXiv:2407.11005*.

Ashok K Goel and Lalith Polepeddi. 2018. Jill Watson. *Learning engineering for online education: Theoretical contexts and design-based examples. Routledge*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yi Liu, Lianzhe Huang, Shicheng Li, Sishuo Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. RECALL: A Benchmark for LLMs Robustness against External Counterfactual Knowledge. *arXiv preprint arXiv:2311.08147*.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2024. CRUD-RAG: A Comprehensive Chinese Benchmark for Retrieval-Augmented Generation of Large Language Models. *arXiv preprint arXiv:2401.17043*.

Subash Neupane, Elias Hossain, Jason Keith, Himanshu Tripathi, Farbod Ghiasi, Noorbakhsh Amiri Golilarz, Amin Amirlatifi, Sudip Mittal, and Shahram Rahimi. 2024. From Questions to Insightful Answers: Building an Informed Chatbot for University Resources. *Preprint*, arXiv:2405.08120.

Keivalya Pandya and Mehfuza Holia. 2023. Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations. *arXiv preprint arXiv:2310.05421*.

Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries. *arXiv preprint arXiv:2401.15391*.

Kevin Wang and Ramon Lawrence. 2024. HelpMe: Student Help Seeking using Office Hours and Email. In *55th ACM Technical Symposium on Computer Science Education V. 1*, pages 1388–1394.

Kevin Wang, Jason Ramos, and Ramon Lawrence. 2023. ChatEd: A Chatbot Leveraging ChatGPT for an Enhanced Learning Experience in Higher Education. *arXiv preprint arXiv:2401.00052*.

Shuting Wang, Jiongnan Liu Shiren Song, Jiehan Cheng, Yuqi Fu, Peidong Guo, Kun Fang, Yutao Zhu, and Zhicheng Dou. 2024. DomainRAG: A Chinese Benchmark for Evaluating Domain-specific Retrieval-Augmented Generation. *arXiv preprint arXiv:2406.05654*.

Di Wu, Wasi Uddin Ahmad, Dejiao Zhang, Murali Krishna Ramanathan, and Xiaofei Ma. 2024. REPOFORMER: Selective retrieval for repository-level code completion. *arXiv preprint arXiv:2403.10059*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, et al. 2024. CRAG–Comprehensive RAG Benchmark. *arXiv preprint arXiv:2406.04744*.

Yiyun Zhao, Prateek Singh, Hanoz Bhathena, Bernardo Ramos, Aviral Joshi, Swaroop Gadiyaram, and Saket Sharma. 2024. Optimizing LLM Based Retrieval Augmented Generation Pipelines in the Financial Domain. In *Proc of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 279–294.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36.