



UltraRAG: A Modular and Automated Toolkit for Adaptive Retrieval-Augmented Generation

Yuxuan Chen^{1*}, Dewen Guo^{1*}, Sen Mei^{2*}, Xinze Li^{2*}, Hao Chen³, Yishan Li¹, Yixuan Wang³, Chaoyue Tang¹, Ruobing Wang⁴, Dingjun Wu¹, Yukun Yan^{3†}

Zhenghao Liu^{2†}, Shi Yu³, Zhiyuan Liu³, Maosong Sun³

¹ModelBest Inc., ²Northeastern University, ³Tsinghua University

⁴University of Chinese Academy of Sciences

Abstract

Retrieval-Augmented Generation (RAG) significantly enhances the performance of large language models (LLMs) in downstream tasks by integrating external knowledge. To facilitate researchers in deploying RAG systems, various RAG toolkits have been introduced. However, many existing RAG toolkits lack support for knowledge adaptation tailored to specific application scenarios. To address this limitation, we propose UltraRAG, a RAG toolkit that automates knowledge adaptation throughout the entire workflow, from data construction and training to evaluation, while ensuring ease of use. UltraRAG features a user-friendly WebUI that streamlines the RAG process, allowing users to build and optimize systems without coding expertise. It supports multimodal input and provides comprehensive tools for managing the knowledge base. With its highly modular architecture, UltraRAG delivers an end-to-end development solution, enabling seamless knowledge adaptation across diverse user scenarios. The code, demonstration videos, and installable package for UltraRAG are publicly available at <https://github.com/OpenBMB/UltraRAG>.

1 Introduction

Large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Guo et al., 2025) have demonstrated impressive capabilities in understanding and reasoning. However, due to the limitations of their parameterized knowledge and hallucinations, LLMs usually generate incorrect responses (Guu et al., 2020; Ji et al., 2023; Xu et al., 2024). To address this, retrieval-augmented generation (RAG) (Lewis et al., 2020; Guu et al., 2020) has emerged as an effective approach that integrates external knowledge sources, enhancing the accuracy and reliability of responses generated by LLMs. Despite its promising potential, RAG still

faces significant challenges in real-world applications. These include the diversity of knowledge corpus formats and modalities (Yu et al., 2024a), the complexity of coordinating multiple components (Li et al., 2024), and the rapid development of algorithms and models. These challenges create significant obstacles for researchers trying to develop RAG systems.

For these reasons, a variety of RAG toolkits have been developed to offer technical support for researchers (Liu, 2022; Chase, 2022; Jin et al., 2024). These tools typically modularize the RAG system (Jin et al., 2024), enabling users to flexibly select and configure different modules, which streamlines both deployment and execution. However, existing RAG toolkits are often overly complex (Chase, 2022) and lack knowledge adaptation designs tailored to real-world requirements (Jin et al., 2024), making it difficult for users to customize and optimize RAG systems for specific scenarios, such as finance and law.

In this paper, we propose UltraRAG, a modular and automated toolkit for adaptive retrieval-augmented generation, enabling users not only to easily deploy and execute RAG systems but also to enhance the RAG pipeline through knowledge adaptation for different scenarios. UltraRAG consists of two global setting modules (Model Management and Knowledge Management) and three core functional modules (Data construction, Training, and Evaluation & Inference), covering all essential components of the RAG pipeline. From knowledge base preparation to automated data generation, model fine-tuning, and comprehensive evaluation, UltraRAG streamlines the full lifecycle of RAG system development. In summary, UltraRAG provides an end-to-end development platform for RAG systems, facilitating rapid system building, scalable deployment, and fair evaluation. The key features of UltraRAG are outlined in Table 1, which includes:

* Equal Contribution.

† Corresponding Authors.

Toolkit	WebUI	Multimodal	Knowledge Management	End-to-End Development	Knowledge Adaptation
LangChain (Chase, 2022)	✗	✓	✓	✗	✗
LlamaIndex (Liu, 2022)	✗	✓	✓	✗	✗
XRAG (Mao et al., 2024)	✓	✗	✗	✗	✗
FastRAG (Abane et al., 2024)	✗	✗	✗	✗	✗
RAGLab (Zhang et al., 2024)	✗	✗	✓	✗	✗
LocalRQA (Yu et al., 2024b)	✓	✗	✗	✓	✗
FlashRAG (Jin et al., 2024)	✓	✓	✗	✓	✗
UltraRAG (Ours)	✓	✓	✓	✓	✓

Table 1: Comparison of UltraRAG Features with Other RAG Frameworks.

User-Friendly WebUI. UltraRAG provides an intuitive WebUI that allows users to easily deploy RAG systems and efficiently process knowledge bases, including encoding and indexing documents in various formats such as TXT, PDF, and Markdown. This user-friendly interface significantly lowers the barrier to usage, allowing individuals with limited technical expertise to quickly build and deploy RAG applications, while reducing both the learning curve and operational complexity.

Multimodal. UltraRAG supports multimodal RAG research and deployment by integrating MLLMs like MiniCPM-V (Yao et al., 2024) and multimodal retrievers (Radford et al., 2021; Zhou et al., 2024). It also incorporates VisRAG (Yu et al., 2024a), a model tailored for domain-specific multimodal scenarios, offering comprehensive technical support.

Knowledge Management. UltraRAG enables parameterized knowledge base management, transforming complex processing into simple configurations. Unlike previous methods (Liu, 2022; Chase, 2022) that impose format and specification constraints, UltraRAG supports diverse document formats, simplifying knowledge base processing.

End-to-End Development. UltraRAG offers an end-to-end RAG solution that covers the entire pipeline, from data construction, model fine-tuning to inference and evaluation. It integrates advanced RAG algorithms (Li et al., 2024; Zeng et al., 2024; Yu et al., 2024a), allowing users to freely combine various techniques and explore numerous configuration possibilities. In addition, UltraRAG includes over 40 widely used datasets, standardized retrieval and generation metrics, and a unified data format.

Knowledge Adaptation. UltraRAG simplifies the knowledge adaptation process by allowing users to provide only domain-specific corpus.

Through its data construction module, the framework automatically generates optimized training data for the entire pipeline, ensuring that the retrieval and generation components are fine-tuned for the specific domain. Our experimental results demonstrate that by adapting knowledge to a particular domain, UltraRAG significantly improves performance, highlighting the advantages of its knowledge adaptation capabilities in real-world applications.

2 Related Work

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) is an effective method for mitigating hallucination and factual inaccuracy issues of large language models (LLMs) (Jiang et al., 2023; Xu et al., 2023; Luo et al., 2023; Hu et al., 2023). It has been widely adopted in various natural language processing (NLP) tasks, such as open-domain QA (Trivedi et al., 2023), language modeling (He et al., 2021), and dialogue (Cai et al., 2019). A typical RAG system consists of two key components, a retriever and a generator (Shi et al., 2023; Yu et al., 2023). The retriever retrieves relevant documents from an external corpus based on the user’s query (Karpukhin et al., 2020; Xiong et al., 2021) and the generator utilizes these documents as context of inputs to augment the generation process (Ram et al., 2023; Xu et al., 2023).

With the continuous advancement of research on RAG systems, recent studies have introduced additional modules and explored various training methods specifically tailored for RAG systems (Yan et al., 2024; Lin et al., 2023; Wang et al., 2023; Wei et al., 2024). For example, Yan et al. (2024) propose an additional retrieval evaluator to refine the quality of retrieved documents. Li et al. (2024)

utilize a rollout method to obtain rewards from the entire RAG system for each module and optimize them based on the reward. The success of these approaches highlights the growing need for a general-purpose RAG toolkit, which can streamline development and evaluation across diverse RAG frameworks.

RAG Toolkits. Various RAG toolkits have been developed to assist users in building customized RAG systems, such as LangChain (Chase, 2022) and LlamaIndex (Liu, 2022). These frameworks modularize RAG pipelines and offer seamless integration with knowledge databases, embedding models, and LLM APIs, thereby streamlining development workflows and broadening their range of applications (deepset.ai, 2023). However, most existing toolkits lack user-friendly WebUIs, do not offer free access to commonly used retrieval corpora, and tend to be overly encapsulated. These limitations significantly hinder their usability and scalability, making them less suitable for both research and practical deployment scenarios (Mao et al., 2024; Jin et al., 2024).

To address these limitations, recent work has introduced more transparent and adaptable RAG toolkits. For example, FastRAG (Abane et al., 2024) is built upon Haystack’s API, allowing users to freely assemble different modules within RAG pipelines. RAGLAB (Zhang et al., 2024) focuses on training RAG systems, offering training strategies tailored for different components. However, these toolkits do not adequately support users in end-to-end deployment and development and are not applicable to multimodal tasks. FlashRAG (Jin et al., 2024) not only addresses several of these challenges but also integrates multiple algorithms, allowing users to efficiently reproduce existing methods and explore novel approaches. However, FlashRAG lacks evaluation for different modules in the RAG system (Mao et al., 2024) and doesn’t support knowledge adaptation for specific scenarios and tasks. In contrast, our proposed UltraRAG toolkit offers an end-to-end modular framework for constructing RAG systems, featuring comprehensive knowledge management and fine-grained evaluation for each module. UltraRAG supports both text and multi-modal tasks, facilitating end-to-end development, evaluation, and deployment of RAG applications.

3 UltraRAG

This section introduces UltraRAG, a modular RAG framework designed for the rapid implementation and deployment of RAG pipelines. As shown in Figure 1, UltraRAG consists of two global setting modules: Model Management and Knowledge Management, along with three functional modules: Data Construction, Training, and Evaluation & Inference. Additionally, UltraRAG offers a user-friendly, visual WebUI, allowing users lacking coding experience to easily process knowledge bases, fine-tune models, and utilize them seamlessly. Details and screenshots of the WebUI can be found in Appendix A.2.

3.1 Global Setting

In this subsection, we introduce the global settings applied in UltraRAG.

Model Management. UltraRAG provides an efficient model management module for the RAG system, enabling the management, deployment, and usage of various models, including retrieval, reranker, and generation models. It supports local models deployed via vLLM (Kwon et al., 2023) or HuggingFace Transformers (Wolf et al., 2020), as well as API-based models. To reduce the learning curve, we provide a pre-configured Docker environment and microservices, enabling models to be preloaded in the background for seamless integration with other modules.

Knowledge Management. External knowledge is fundamental to RAG systems, as it provides the documents available for retrieval. However, managing knowledge bases can be challenging, especially for beginners. To address this, UltraRAG offers a user-friendly knowledge management module that enables users to upload knowledge base files in various formats, such as JSON and CSV, and deploy them seamlessly. In addition, users can adjust parameters within the knowledge management module to process the deployed knowledge base, such as adjusting the chunk size, configuring the overlap between chunks, and selecting the embedding model for encoding. Once processed, users can easily obtain the encoded knowledge base along with its corresponding search indexes.

3.2 Functional Modules

In this subsection, we introduce several functional modules implemented in UltraRAG, covering the entire workflow of a RAG system.

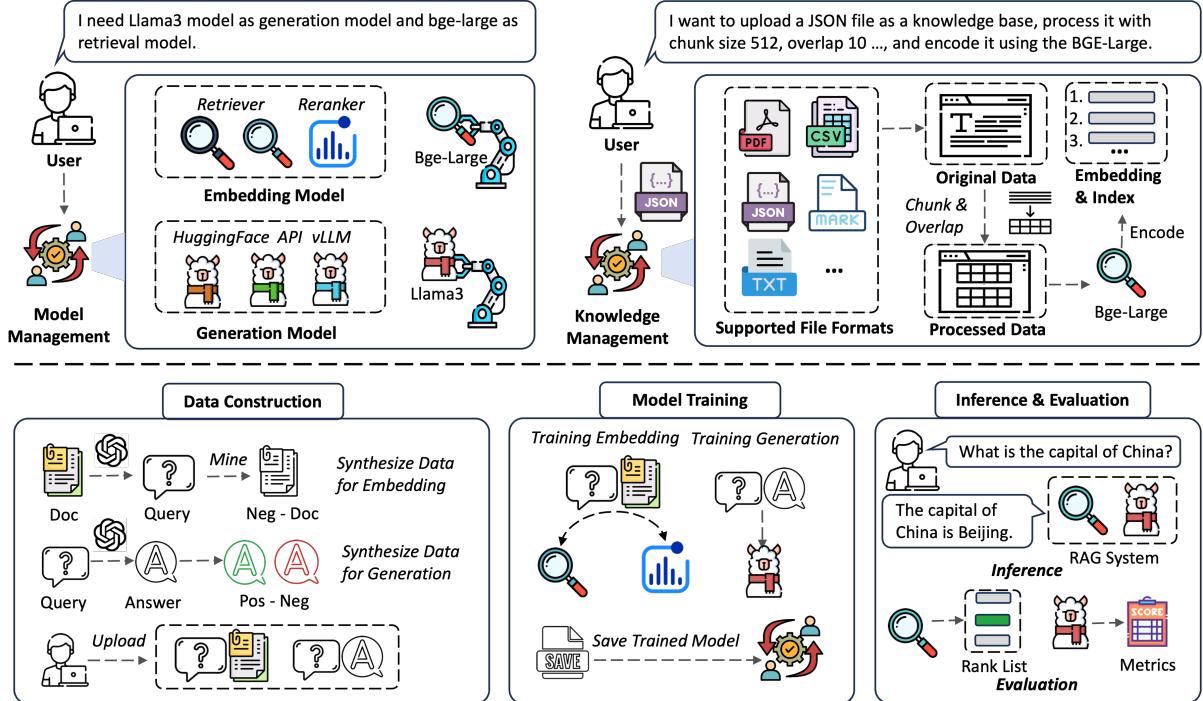


Figure 1: The Overall Architecture of UltraRAG Framework.

Data Construction. The data construction module integrates advanced data synthesis techniques (Zhu et al., 2024; Li et al., 2024; Zeng et al., 2024) to provide datasets to train and evaluate different models within RAG systems. UltraRAG first generates queries automatically based on documents in the user-provided knowledge base. These queries are then used to construct training and evaluation datasets for both retrieval and generation models. For retrieval and reranking models, UltraRAG synthesizes query-document pairs and mines hard negative samples for each query (Xiong et al., 2020). For generation models, it builds on prior work (Li et al., 2024) to construct supervised fine-tuning (SFT) datasets and direct preference optimization (DPO) datasets, where each query serves as the input for generating high-quality responses. Additionally, users can upload their own pre-constructed datasets and adjust the data proportions to mix different datasets, enabling more effective multi-task training.

Training. Users can further enhance downstream task performance through fine-tuning. Leveraging the training data provided by the data construction module, the training module supports fine-tuning for both embedding models and generation models. Currently, UltraRAG implements two alignment strategies: supervised fine-tuning (SFT) and direct preference optimization (DPO),

with plans to incorporate more training strategies in future updates.

Evaluation & Inference. UltraRAG’s evaluation module provides users with comprehensive methods to assess the performance of both embedding and generation models. It supports a wide range of commonly used retrieval and generation metrics and provides access to over 40 benchmark datasets. In addition, UltraRAG defines a unified data format and allows users to add custom datasets, facilitating flexible and extensible evaluation workflows.

The inference module is available for users who require direct access to RAG services, offering four predefined workflows: Vanilla RAG, KBAlign (Zeng et al., 2024), VisRAG (Yu et al., 2024a), and Adaptive-Note (Wang et al., 2024), while also allowing users to create custom inference workflows according to their specific needs. In addition to these workflows, the module supports streaming outputs and visualizes intermediate retrieval and reasoning processes, enabling users to monitor the performance of both retrieval and generation models in real time. Furthermore, UltraRAG supports local deployment via Ollama¹, allowing users to deploy models directly on the WebUI as RAG applications that can be fully customized to their specific knowledge bases.

¹<https://ollama.com/>

Feature	Model
Multimodal	VisRAG
Knowledge Management	Adaptive-Note
End-to-End Development	RAG-DDR, RA-DIT, KBAlign
Knowledge Adaption	RAG-DDR

Table 2: Typical Implementation. The different features of UltraRAG can support various models and methods.

3.3 Typical Implementation

UltraRAG is designed to simplify the research process for researchers by eliminating the need for repetitive development of RAG’s modular components. As shown in Table 2, we have already implemented several RAG methods, including Vanilla RAG, RA-DIT (Lin et al., 2023), Adaptive-Note (Wang et al., 2024), VisRAG (Yu et al., 2024a), KBAlign (Zeng et al., 2024), and RAG-DDR (Li et al., 2024). Looking ahead, we plan to expand the framework by incorporating additional RAG baselines. Given the rapid evolution of RAG and the lack of standardized evaluation methods, we believe UltraRAG will accelerate the reproduction of results and enable fair comparisons across different RAG baselines.

4 Knowledge Adaptation for Legal-Domain RAG System with UltraRAG

In this section, we present the development of a legal-domain RAG system to highlight UltraRAG’s knowledge adaptation capabilities. We choose the legal domain because the complexity of legal terminology poses significant challenges for LLMs, making domain-specific adaptation essential for improving their real-world applicability.

4.1 Preliminary of Legal-Domain RAG System

This subsection introduces the evaluation benchmark, models, and knowledge base used in our experiments. Leveraging UltraRAG’s model and knowledge management modules, we complete the setup effortlessly without coding.

Evaluation Dataset. We select LawBench as the evaluation scenario, a meticulously designed dataset for assessing the legal capabilities of large models within the context of the Chinese legal system. We choose the Scene-based Article Prediction (3-2) and Consultation (3-8) tasks for evaluation. The Scene-based Article Prediction task requires

the model to predict relevant legal provisions based on a given scenario and question. The Consultation task focuses on generating appropriate responses to user legal consultations, emphasizing the provision of legal advice based on relevant legal provisions. We choose these tasks because they are more closely aligned with real-world business scenarios.

Models. We use MiniCPM-Embedding-Light² as the embedding model and MiniCPM-3-4B³ as the generation model. These local models are loaded through the model management module to ensure efficient invocation.

Knowledge Base. We have collected a comprehensive knowledge base consisting of over 1,000 law-related books, covering topics such as civil law, criminal law, and judicial cases, to support RAG applications in the legal domain. To optimize retrieval performance and ensure ease of use, we upload all files through the knowledge management module. The index is built by setting the chunk size to 512 with a 15% overlap, enhancing the retrieval accuracy.

4.2 Methodology Powered by UltraRAG

In this subsection, we introduce the methods provided by UltraRAG, which enhance model adaptability and performance in downstream domains through self-constructed training data and model training.

Embedding Finetuning. To finetune the embedding models, we utilize the data construction module of UltraRAG to synthetically generate 2,800 finetuning samples, improving the performance of the MiniCPM-Embedding-Light model. This finetuning process allows the model to better adapt to domain-specific tasks, boosting its retrieval accuracy. The model excels in cross-lingual retrieval, supporting both Chinese and English, and generates high-quality embeddings through bidirectional attention mechanisms and weighted mean pooling. It is also capable of handling long texts of up to 8,192 tokens. To further investigate the training benefits of UltraRAG for the generation models, we use the vanilla MiniCPM-Embedding-Light as the retriever for all the following baselines, and finetune the MiniCPM-3-4B model.

Generation Finetuning. UltraRAG integrates two training methods, UltraRAG-DDR and UltraRAG-KBAlign, to train the generation model.

²<https://huggingface.co/openbmb/MiniCPM-Embedding-Light>

³<https://huggingface.co/openbmb/MiniCPM3-4B>

Method	MRR@10	NDCG@10	Recall@10
UltraRAG-Embedding	36.46	40.05	54.50
w. Finetune	37.57	42.12	56.50

Table 3: Retrieval Performance of UltraRAG with Knowledge Adaptation.

UltraRAG-DDR. We implement UltraRAG-DDR following Li et al. (2024). Leveraging the DDR method provided by UltraRAG’s data construction module, we generate training data by constructing query, ground-truth, and keypoint triplets, and apply data sampling strategies to create diverse responses. Using rule-based rewards, we construct preference pairs, and finally, finetune the model using DPO loss (Rafailov et al., 2023) with LoRA (Hu et al., 2022).

UltraRAG-KBAlign. We follow the approach in Zeng et al. (2024) and apply LoRA-based SFT for finetuning, enhancing the model’s adaptability to the knowledge base through self-supervised learning. The data construction process combines short-range annotations (local information within a single chunk) and long-range annotations (cross-chapter information integration) to improve the model’s knowledge integration capability. Ultimately, during the inference phase, we optimize the aligned knowledge representations to enhance the accuracy and consistency of the generated answers.

Inference Workflow. UltraRAG integrates three different RAG inference workflow.

UltraRAG-VanillaRAG. This is a basic RAG workflow, where user task requirements are fulfilled using vanilla retriever and generation model.

UltraRAG-DeepNote. We implement the RAG workflow based on Wang et al. (2024), which uses an adaptive memory reviewer and a stop-exploration strategy to iteratively collect and update knowledge, enhancing both information retrieval and generation quality, and enabling adaptation to more complex user tasks.

UltraRAG-RAGAdaptation. In contrast to the previous two workflows, this framework represents the use of models finetuned by the user for knowledge adaptation in specific scenarios. Here, we employ the retrieval and generation models trained with UltraRAG-Embedding and UltraRAG-DDR.

Law Prediction (3-2)		Consultation (3-8)	
Method	ROUGE-L	Method	ROUGE-L
VanillaRAG	40.75	VanillaRAG	23.65
KBAlign	48.72	DeepNote	24.62
DDR	53.14	RAGAdaptation	25.85

Table 4: Generation Performance of UltraRAG with Knowledge Adaptation.

4.3 UltraRAG Performance in Legal Scenarios

In this experiment, we explore the effectiveness of knowledge adaptation in legal scenarios enabled by the UltraRAG training module and assess its effectiveness using the evaluation module. Additionally, we provide case studies in Appendix A.1.

For retrieval evaluation, we use MRR@10, NDCG@10, and Recall@10, assessing MiniCPM-Embedding-Light on 200 GPT-4o-annotated samples before and after finetuning. As shown in Table 3, after finetuning on domain-specific data, the performance of the retriever has improved. Through knowledge adaptation in legal scenarios, the retriever better captures the correlation between queries and legal documents, enabling more precise retrieval.

Next, we explore the performance of the generation model and compare different inference workflows. As shown in Table 4, both KBAlign and DDR demonstrate performance gains, with DDR achieving a 30% relative improvement. Compared to VanillaRAG, DeepNote significantly enhances performance through its unique memory mechanism, effectively structuring external knowledge. Meanwhile, RAGAdaptation achieves the best results, enabling users to customize and finetune models for specific domains. These findings further highlight the importance of knowledge adaptation and the potential of the UltraRAG framework.

5 Conclusion

In this paper, we present UltraRAG, a novel framework designed to address the challenges of domain-specific knowledge adaptation in Retrieval-Augmented Generation systems. With its comprehensive workflow, extensible architecture, and user-friendly WebUI, UltraRAG greatly reduces the barrier to usage, shortens the learning curve, and offers flexible module combinations along with typical implementations, making both the usage and development of RAG systems more adaptable.

References

- Amar Abane, Anis Bekri, and Abdella Battou. 2024. Fastrag: Retrieval augmented generation for semi-structured data. *arXiv preprint arXiv:2411.13773*.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019. Skeleton-to-response: Dialogue generation guided by retrieval memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1219–1228.
- Harrison Chase. 2022. [LangChain](#).
- deepset.ai. 2023. [Haystack](#).
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714.
- Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *ArXiv preprint*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of EMNLP*, pages 7969–7992.
- Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *arXiv preprint arXiv:2405.13576*.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- 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.
- Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, et al. 2024. Rag-ddr: Optimizing retrieval-augmented generation using differentiable data rewards. *arXiv preprint arXiv:2410.13509*.
- Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvassy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *ArXiv preprint*.
- Jerry Liu. 2022. [LlamaIndex](#).
- Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *ArXiv preprint*.
- Qianren Mao, Yangyifei Luo, Jinlong Zhang, Hanwen Hao, Zhilong Cao, Xiaolong Wang, Xiao Guan, Zhenting Huang, Weifeng Jiang, Shuyu Guo, et al. 2024. Xrag: examining the core-benchmarking foundational components in advanced retrieval-augmented generation. *arXiv preprint arXiv:2412.15529*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, pages 1316–1331.

- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. [Replug: Retrieval-augmented black-box language models](#). *ArXiv preprint*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.
- Ruobing Wang, Daren Zha, Shi Yu, Qingfei Zhao, Yuxuan Chen, Yixuan Wang, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, et al. 2024. Retriever-and-memory: Towards adaptive note-enhanced retrieval-augmented generation. *arXiv preprint arXiv:2410.08821*.
- Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.
- Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation with explicit denoising. *arXiv e-prints*, pages arXiv–2406.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). In *Proceedings of ICLR*.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. [Recamp: Improving retrieval-augmented lms with compression and selective augmentation](#). *ArXiv preprint*.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective retrieval augmented generation](#).
- Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.
- Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. 2024a. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*.
- Xiao Yu, Yunan Lu, and Zhou Yu. 2024b. Localrqa: From generating data to locally training, testing, and deploying retrieval-augmented qa systems. *arXiv preprint arXiv:2403.00982*.
- Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. [Augmentation-adapted retriever improves generalization of language models as generic plug-in](#). *ArXiv preprint*.
- Zheni Zeng, Yuxuan Chen, Shi Yu, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Kbalign: Kbalign: Efficient self adaptation on specific knowledge bases. *arXiv preprint arXiv:2411.14790*.
- Xuanwang Zhang, Yunze Song, Yidong Wang, Shuyun Tang, Xinfeng Li, Zhengran Zeng, Zhen Wu, Wei Ye, Wenyuan Xu, Yue Zhang, et al. 2024. Raglab: A modular and research-oriented unified framework for retrieval-augmented generation. *arXiv preprint arXiv:2408.11381*.
- Tianshuo Zhou, Sen Mei, Xinze Li, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, Yu Gu, and Ge Yu. 2024. Marvel: Unlocking the multi-modal capability of dense retrieval via visual module plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14608–14624.
- Kunlun Zhu, Yifan Luo, Dingling Xu, Ruobing Wang, Shi Yu, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, et al. 2024. Rageval: Scenario specific rag evaluation dataset generation framework. *arXiv preprint arXiv:2408.01262*.

A Appendix

A.1 Case Study

In Table 5, we present two cases to further demonstrate the effectiveness of UltraRAG’s knowledge adaptation.

As shown in the first case, which addresses labor protection issues in foreign-invested enterprises, VanillaRAG mistakenly references Article 38 of the Labor Law. However, the core issue relates to the responsibilities of foreign-invested enterprises, not general labor law. After being fine-tuned for legal domain-specific tasks, UltraRAG accurately understands the issue and correctly cites Article 32 of the Administrative Regulations on Foreign-invested Enterprises, providing the appropriate response.

In the second case, both models offer a correct response. However, VanillaRAG simply states “auction concluded” without referencing the specific legal provision. In contrast, UltraRAG delivers a more accurate response by citing the relevant legal article, better addressing the user’s needs.

A.2 Screenshots of UltraRAG

In this subsection, we begin by introducing the WebUI interface of UltraRAG, highlighting its key features and functionalities, followed by a detailed description of the DeepNote workflow.

As shown in Figure 2(a), the global settings interface enables seamless navigation between the model management and knowledge management modules. Within the model management module, users can easily select the desired model, specify the model path, assign a CUDA device, and configure other essential settings. These features streamline the deployment process, allowing users to quickly set up a RAG system.

Next, we describe the workflow of UltraRAG DeepNote. As illustrated in Figure 2(b), UltraRAG DeepNote introduces an advanced memory mechanism that dynamically organizes and updates the retrieved knowledge. Unlike VanillaRAG, which directly concatenates retrieved documents to the context, UltraRAG DeepNote continuously organizes knowledge throughout the inference process. This approach enhances knowledge structuring and improves information integration, leading to more accurate and coherent responses.

Case #1

Instruction: Please provide the legal basis according to the specific scenario and issue, only including the relevant article of law, with each scenario involving only one article.

Query: Scenario: Xiao Ming works at a foreign-invested enterprise and suffers a work injury due to an unsafe working environment. According to which law should the foreign-invested enterprise comply with labor protection regulations?

Ground Truth: According to Article 32 of the Foreign Investment Law, foreign-invested enterprises shall comply with the relevant labor protection provisions of laws and administrative regulations when carrying out production and business activities. Therefore, foreign-invested enterprises should ensure a safe working environment for employees and comply with national labor protection laws and regulations. When employees are injured due to work-related reasons, the enterprise should provide appropriate medical treatment and actively assist employees in applying for work-related injury insurance benefits.

VanillaRAG Predit: According to Article 38 of the Labor Law of the People’s Republic of China, workers have the right to social insurance and benefits. Workers are entitled to medical leave if they are ill or injured not due to work, and during the treatment for work-related injuries, their original salary should remain unchanged. The employer should pay the relevant benefits based on the principle of work-related injury.

UltraRAG-RAGAdaption Predit: According to Article 32 of the Regulations on the Administration of Foreign-invested Enterprises, foreign-invested enterprises must comply with the relevant laws and administrative regulations concerning labor protection and social insurance. Therefore, foreign-invested enterprises should adhere to labor protection regulations, ensure a safe working environment, and prevent work-related accidents.

Case #2

Instruction: Please provide the specific scenario and question, and I will give you the relevant legal provision. Each scenario will be linked to a single legal article.

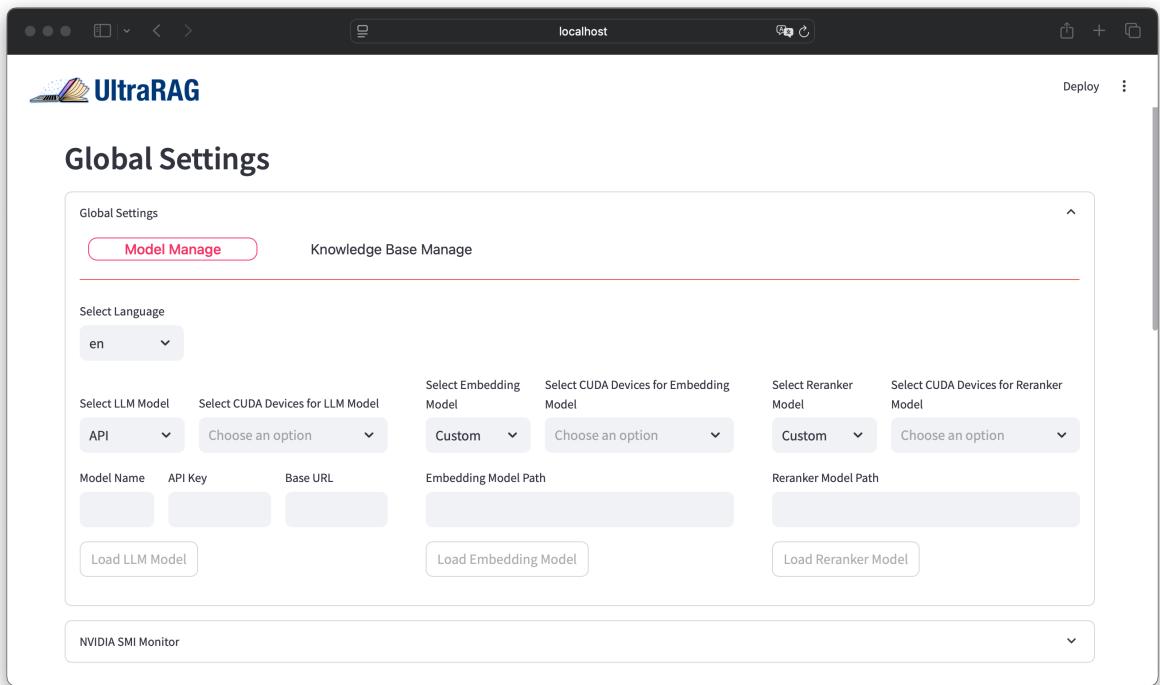
Query: In an auction, a bidder bids a high price for a painting, and the auctioneer confirms the highest bid by dropping the hammer. According to the law, what is the status of the auction once the bidder’s highest offer is confirmed?

Ground Truth: According to Article 51 of the Auction Law, once the highest bid of the bidder is confirmed by the auctioneer’s hammer or other public methods of indicating acceptance, the auction is concluded. Therefore, in this scenario, after the auctioneer’s confirmation, the bidder’s highest offer is finalized and publicly announced, and the auction is deemed to have been successfully concluded.

VanillaRAG Predit: The auction is concluded.

UltraRAG-RAGAdaption Predit: According to Article 51 of the Auction Law, once the highest bid of the bidder is confirmed by the auctioneer’s hammer or other public means of indicating that the sale is final, the status of the auction is “auction concluded”.

Table 5: Case Studies. Different colors are used to annotate these matched phrases for each model: Blue for Ground Truth, Orange for Vanilla RAG and Green for RAGAdaption.



(a) Global Setting.

This screenshot shows the 'DeepNote' interface of UltraRAG. At the top, there is a message from a user asking about legal liabilities after hitting someone while intoxicated.

The AI response is as follows:

► Note Init: What liabilities might one bear after hitting someone while intoxicated?
 ► Note Update: 1. What specific forms of liability or responsibility may result from causing harm while intoxicated? 2. Under what circumstances might the liability of an intoxicated individual be reduced or exempted in cases of causing harm?
 ► Note Update: 1. How can one prove the causation between voluntarily intoxicated behavior and resulting harm in legal proceedings? 2. Are there specific legal provisions or precedents regarding the criminal liability of intoxicated drivers?
 ► Note Summary:
 ► thinking
 After hitting someone while intoxicated, the individual may bear the following liabilities:

1. **Criminal Liability:** The person could be charged with a crime, such as drunk driving (危险驾驶罪) or traffic manslaughter (交通肇事罪), potentially leading to imprisonment.
2. **Civil Liability:** The individual may be required to compensate the victim for damages, including medical expenses, lost wages, and pain and suffering.
3. **Administrative Liability:** The person might face administrative penalties, such as吊销驾驶执照 (having their driver's license revoked) and fines.
4. **Compensation Obligation:** Even if intoxicated, the individual may need to compensate the victim based on their economic situation, as per the侵权责任法 (In Tort Liability Law).

In summary, the intoxicated person may incur criminal, civil, and administrative responsibilities, along with the obligation to compensate the victim for their losses.
 spend: 88.61 s, first response 78.34 s

A text input field at the bottom is labeled 'Please input your question'.

(b) UltraRAG DeepNote.

Figure 2: Screenshots of UltraRAG.