

# RETA-LLM: A Retrieval-Augmented Large Language Model Toolkit

Jiongnan Liu<sup>1</sup>, Jiajie Jin<sup>2</sup>, Zihan Wang<sup>1</sup>, Jiehan Cheng<sup>1</sup>, Zhicheng Dou<sup>1\*</sup>, and Ji-Rong Wen<sup>1</sup>

<sup>1</sup>Gaoling School of Artificial Intelligence, Renmin University of China

<sup>2</sup>University of Science and Technology of China

<sup>1</sup>{liujn, wangzihan0527, jiehan\_cheng, dou, jrwen}@ruc.edu.cn

<sup>2</sup>jinjiajie@mail.ustc.edu.cn

## Abstract

Although Large Language Models (LLMs) have demonstrated extraordinary capabilities in many domains, they still have a tendency to hallucinate and generate fictitious responses to user requests. This problem can be alleviated by augmenting LLMs with information retrieval (IR) systems (also known as retrieval-augmented LLMs). Applying this strategy, LLMs can generate more factual texts in response to user input according to the relevant content retrieved by IR systems from external corpora as references. In addition, by incorporating external knowledge, retrieval-augmented LLMs can answer in-domain questions that cannot be answered by solely relying on the world knowledge stored in parameters. To support research in this area and facilitate the development of retrieval-augmented LLM systems, we develop RETA-LLM, a RETreival-Augmented LLM toolkit. In RETA-LLM, we create a complete pipeline to help researchers and users build their customized in-domain LLM-based systems. Compared with previous retrieval-augmented LLM systems, RETA-LLM provides more plug-and-play modules to support better interaction between IR systems and LLMs, including request rewriting, document retrieval, passage extraction, answer generation, and fact checking modules. Our toolkit is publicly available at <https://github.com/RUC-GSAI/YuLan-IR/tree/main/RETA-LLM>.

## 1 Introduction

Large language models (LLMs) have attracted increasing attention from both research community and industry (Brown et al., 2020; OpenAI, 2023; Ouyang et al., 2022; Touvron et al., 2023; Chowdhery et al., 2022; Zhao et al., 2023; Zeng et al., 2022). With tremendous world knowledge stored in parameters (Petroni et al., 2019; Roberts et al., 2020; Jiang et al., 2020) and the Reinforcement Learning

from Human Feedback (RLHF) techniques (Christiano et al., 2017; Ziegler et al., 2019), LLMs can generate helpful, detailed, and polite texts in response to user inputs. Many studies have demonstrated LLMs’ extraordinary abilities in various areas, including nature language processing (Moslem et al., 2023), information retrieval (Sun et al., 2023; Wang et al., 2023; Mao et al., 2023), and recommendation (Hou et al., 2023; Zhang et al., 2023).

However, LLMs still tend to hallucinate and sometimes generate texts opposite to facts (Zhou et al., 2021; Zhao et al., 2023). To tackle these problems, researchers have proposed a new paradigm to strengthen LLMs with information retrieval systems (retrieval-augmented LLMs) (Shi et al., 2023; Jiang et al., 2023; Nakano et al., 2022), which enables LLMs to retrieve relevant contents from an external repository (knowledge corpus) to generate texts based on them. It has been verified that retrieval-augmented LLMs can generate texts in response to user input with fewer hallucinations (Nakano et al., 2022). Furthermore, by incorporating customized private data resources, retrieval-augmented LLMs can respond to in-domain queries that cannot be answered by LLMs trained with public data.

To support research in this area and help users build their own in-domain LLM-based systems, we devise RETA-LLM, a RETreival-Augmented LLM toolkit. Different from previous general LLM-enhanced toolkits such as LangChain,<sup>1</sup> RETA-LLM focuses on the retrieval-augmented LLMs and provides more plug-in modules. Typically, retrieval-augmented LLMs use a retrieve-and-generate strategy with two modules: First, they retrieve documents or passages based on user request (**document retrieval** module); then, they generate answers utilizing these relevant documents as references (**answer generation** module). In addition,

\*Corresponding author.

<sup>1</sup>LangChain, <https://github.com/hwchase17/langchain>

<https://github.com/hwchase17/langchain>

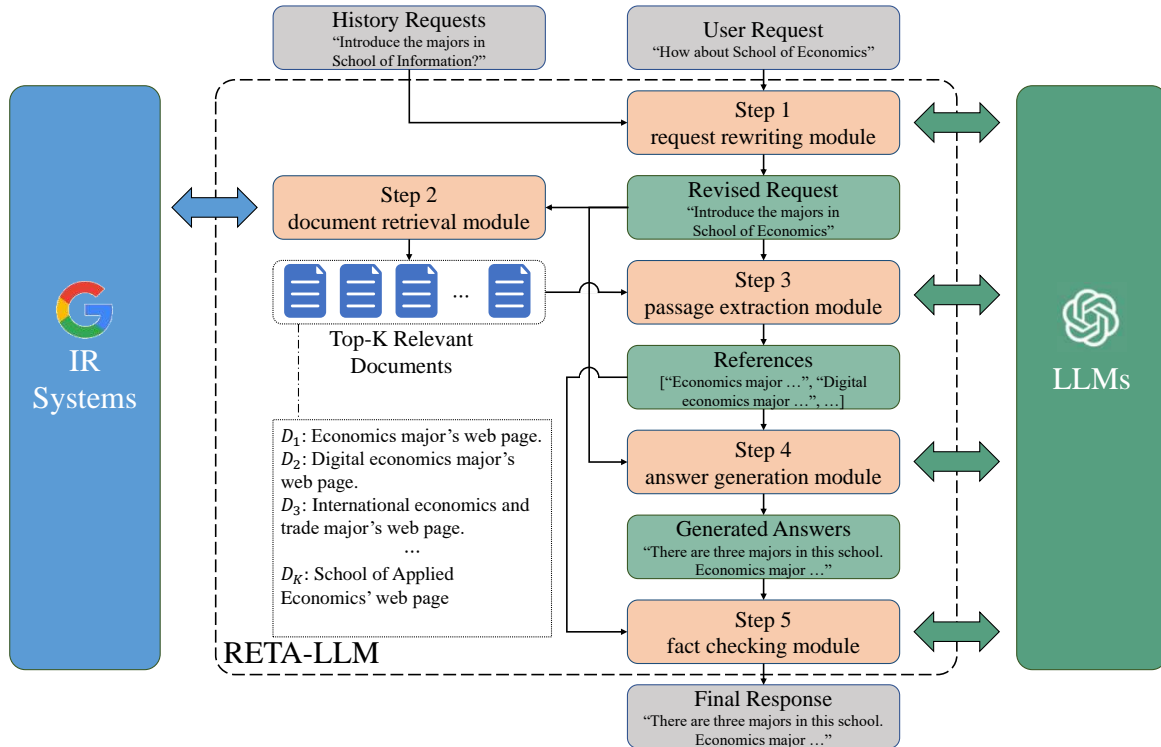


Figure 1: The RETA-LLM framework. Examples are taken from an intelligent university information seeking system powered by RETA-LLM.

tion to these two basic modules, our RETA-LLM provides three optional modules: (1) a **request rewriting** module to make user’s current request more complete and clear; (2) a **passage extraction** module to extract relevant passages or fragments from the whole retrieved document contents; and (3) a **fact checking** module to verify whether there exist factual errors in the generated answers. These optional modules can make the interaction between IR systems and LLMs more effective and smooth. The disentanglement between LLMs and IR systems in our RETA-LLM is more thorough, which makes the customization of search engines and LLMs more convenient. Furthermore, to make the usage easier, we provide a complete and ready-to-use pipeline for researchers and users to build their RETA-LLM toolkits based on their own repository for in-domain LLM-based systems from scratch.

RETA-LLM is part of YuLan, a open source LLM initiative proposed by Gaoling School of Artificial Intelligence, Renmin University of China. RETA-LLM is still under development and there are many issues that need to be solved with great efforts. We sincerely welcome contributions on this open source toolkit.

## 2 RETA-LLM Framework

As aforementioned, compared with Langchain, which is a common LLM-augmented toolkit, our RETA-LLM toolkit focuses specifically on retrieval-augmented LLMs. We provide five plug-in modules in RETA-LLM to interact with LLMs and IR systems. The modules include request rewriting, document retrieval, passage extraction, answer generation, and fact checking modules. The framework of our RETA-LLM is shown in Figure 1. The workflow of RETA-LLM is as follows:

First, RETA-LLM uses the request rewriting module to revise the current user request to make it complete and clear. Because users can issue a series of questions to the RETA-LLM, the semantics of the current user request may be incomplete. For example, A user may ask “*How about the School of Economics?*” while the historical request is “*Introduce the majors in School of Information*”. In this case, the precise meaning of the user is “*Introduce the majors in School of Economics*”. Since LLMs have shown remarkable abilities in rewriting queries in conversational dense retrieval (Mao et al., 2023), we feed the current user request and the previous conversation histories to LLMs to perform rewriting.

Then, RETA-LLM uses the document retrieval module to retrieve relevant documents from the external corpus based on the revised user request. The document retrieval module is the module connected to the IR system. It retrieves relevant documents from the external knowledge corpus and returns top- $K$  of them. The  $K$  is set to 3 in our default configuration. We provide a default dense retriever in our repository. The detailed description can be found in the next section.

Next, RETA-LLM uses the passage extraction module to extract fragments related to the user request from the retrieved documents to form the references. Because of the input length limitations (typically 2048 or 4096 tokens) of LLMs, it is impossible to directly concatenate the contents of all top- $K$  relevant document contents as references for them to generate answers. Trivial methods by truncating the document contents may lose important information in them. Therefore, we reuse the LLMs themselves to extract related fragments from retrieved documents based on the revised request. Since the length of one document may also exceed the limitations, we apply the sliding window strategy to extract fragments step by step. The sliding window size and step are set to 512 and 256 in our default configuration. These fragments are then concatenated together as the references.

Besides, RETA-LLM uses the answer generation module to generate answers for the user request. As previous researches (Nakano et al., 2022; Shi et al., 2023; Jiang et al., 2023) suggest, by feeding the references retrieved from the external corpus, LLMs can generate more factual answers.

Finally, RETA-LLM uses the fact checking module to verify whether the generated answers contain factual mistakes and output final responses for the user request. Though providing additional evidence for generation, LLMs may also hallucinate (Nakano et al., 2022). It is necessary to devise a module to conduct further fact verification. Because of the strong natural language understanding abilities of LLMs, we feed the references and generated answers to them to make judgments. Therefore, RETA-LLM can decide whether to output the generated answers or just say “*I cannot answer this question*”.

Noticed that all the inputs to the LLMs are wrapped in instructions or prompts. As shown in Figure 1, we disentangle the IR systems and LLMs entirely in our RETA-LLM. This separate design

in our RETA-LLM leads users can customize their personal search engines and LLMs.

### 3 RETA-LLM Usage Pipeline

To make the toolkit more convenient for personal usage, we provide a complete pipeline to build in-domain LLM-based system based on html resources. The pipeline is as follows:

First, RETA-LLM uses Beautiful Soup package to convert the raw html files into json data in our **HTML Converter**.<sup>2</sup>

Second, RETA-LLM follows the implementation of disentangled-retriever (Zhan et al., 2022) to build dense indexes and to conduct domain adaption from the converted json data in our **Index Builder**.<sup>3</sup> Specifically, our method supports unsupervised training of dense retrieval models on local document collections, enabling the model to learn domain-specific knowledge in advance. Compared with the retrieval module in the popular LangChain library, our retrieval method has two advantages: (1) the model learns knowledge within the domain of local documents, enabling it to match queries more accurately, and (2) our method does not segment text, thus avoiding any negative impact on the overall semantic information of the text. We also provide a sparse retriever applying faiss (Johnson et al., 2019) package to build sparse indexes.<sup>4</sup> Otherwise, users can also use their customized search engines as the document retrieval module.

Third, users need to prepare LLMs for question answering. For LLM loading and responding, we provide the template for Alpaca (Taori et al., 2023),<sup>5</sup> YuLan-Chat,<sup>6</sup> ChatGLM (Zeng et al., 2022; Du et al., 2022),<sup>7</sup> and GPT-3.5 API (Ouyang et al., 2022).<sup>8</sup> If users use other LLMs, they can edit the codes and configurations in our toolkit.

Finally, users can start their own RETA-LLM services using streamlit package.<sup>9</sup>

<sup>2</sup>Beautiful Soup, <https://beautiful-soup-4.readthedocs.io/en/latest/>

<sup>3</sup>disentagled-retriever, <https://github.com/jingtaozhan/disentagled-retriever>

<sup>4</sup>Faiss, <https://github.com/facebookresearch/faiss>

<sup>5</sup>Alpaca, [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca)

<sup>6</sup>YuLan-Chat, <https://github.com/RUC-GSAI/YuLan-Chat>

<sup>7</sup>ChatGLM, <https://github.com/THUDM/ChatGLM-6B>

<sup>8</sup>OpenAI’s API, <https://api.openai.com/v1/completions>

<sup>9</sup>streamlit, <https://github.com/streamlit/streamlit>



介绍一下历史学院的专业

AI is generating response:

历史学院是中国人民大学的一部分，由历史系、考古文博系、清史研究所和民国史研究院组成。该学院以培养具有深厚人文底蕴、世界视野和扎实专业基础的历史专业人才而著称。学院提供了3个本科招生专业：历史学、世界史和考古学，以及9个硕士学科点和8个博士学科点。中国史学科已于2017年被入选为“世界一流学科”建设名单。学院还是国家级人才培养重大项目最多的历史学院系之一，包括“国家级一流本科专业”建设点、“强基计划”和“基础学科拔尖学生培养计划2.0基地”。学院高度重视本科教学改革，建立起全方位人才培养体系，培养通专结合、中西融合跨学科高水平人才。

Reference URL:

- 历史学院: <http://rdzs.ruc.edu.cn/cms/view/college/4/>
- 历史学: <http://rdzs.ruc.edu.cn/cms/view/professional/97/>
- 中国共产党历史: <https://rdzs.ruc.edu.cn/cms/view/enroll/professional/29/>

Figure 2: A case in RUC-enrollment-assistant system.

More details about the usage pipeline can be found on our GitHub repository.

## 4 A RETA-LLM Service Case

Based on the RETA-LLM and the usage pipeline, we use the web pages on Renmin University of China’s enrollment online platform,<sup>10</sup> to build an RUC-enrollment-assistant system. The system uses a dense document retrieval module and adopts YuLan-13B as the backbone LLM. A using case is shown in 2. By enhancing the IR systems, LLMs can answer in-domain questions which cannot be answered by their own knowledge.

## 5 Conclusion and Future Work

In this paper, we propose RETA-LLM to facilitate research and development of retrieval-augmented LLMs. We provide five independent modules: request rewriting, document retrieval, passage extraction, answer generation, and fact checking modules in our toolkit. Furthermore, we provide a pipeline to help users build their in-domain LLM-based systems. In the future, we are going to include more retrieval-augmented LLM strategies such as active retrieval augmented generation (Jiang et al., 2023). Besides, we plan to make RETA-LLM more modularized and configurable.

<sup>10</sup>Renmin University of China’s enrollment online platform, <https://rdzs.ruc.edu.cn>

## References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. *Language models are few-shot learners*. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NIPS*, pages 4299–4307.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. *Large language models are zero-shot rankers for recommender systems*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. *How Can We Know What Language Models Know? Transactions of the Association for Computational Linguistics*, 8:423–438.

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](#).
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Kelong Mao, Zhicheng Dou, Haonan Chen, Fengran Mo, and Hongjin Qian. 2023. [Large language models know your contextual search intent: A prompting framework for conversational search](#).
- Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. [Adaptive machine translation with large language models](#).
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. [Webgpt: Browser-assisted question-answering with human feedback](#).
- OpenAI. 2023. [Gpt-4 technical report](#).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. [How much knowledge can you pack into the parameters of a language model?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- 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](#). *CoRR*, abs/2301.12652.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. [Is chatgpt good at search? investigating large language models as re-ranking agent](#).
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. [Stanford alpaca: An instruction-following llama model](#). [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Liang Wang, Nan Yang, and Furu Wei. 2023. [Query2doc: Query expansion with large language models](#). *CoRR*, abs/2303.07678.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. [Glm-130b: An open bilingual pre-trained model](#). *arXiv preprint arXiv:2210.02414*.
- Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jiabin Mao, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. [Disentangled modeling of domain and relevance for adaptable dense retrieval](#). *arXiv preprint arXiv:2208.05753*.
- Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. [Recommendation as instruction following: A large language model empowered recommendation approach](#). *CoRR*, abs/2305.07001.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. [A survey of large language models](#).
- Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. [Detecting hallucinated content in conditional neural sequence generation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. [Fine-tuning language models from human preferences](#). *CoRR*, abs/1909.08593.