


FedRAG: Federated Retrieval Augmented Generation

Jungwon Seo 

University of Stavanger, Stavanger, Norway
jungwon.seo@uis.no

Abstract. The Retrieval Augmented Generation (RAG) approach offers a straightforward and efficient method for integrating new information into existing large language models (LLMs) without requiring additional training. However, when applying this RAG-based LLM in a distributed data source environment, there arises a need to either centralize the data or make it externally accessible. This poses potential challenges related to data ownership rights and the inconvenience of pre-processing sensitive information. In this research, we propose a novel approach to construct a RAG-based LLM pipeline within a federated environment, thus eliminating the necessity for data transfer. Our solution leverages global-local question-answering technology, an expedited engineering strategy aimed at extracting comprehensive answers from limited data sources.

Keywords: Large Language Models · Retrieval Augmented Generation.

1 Introduction

Large Language Models (LLMs) have recently exhibited impressive performance in various domains of Natural Language Processing (NLP) [1]. This has spurred numerous industries to develop specialized LLM systems tailored to their needs. Additionally, ongoing research is focused on exploring the potential of federated learning to tackle the inherent data privacy and regulatory challenges associated with machine learning. In federated learning, the training of LLMs typically involves fine-tuning and optimizing models for downstream tasks rather than starting from scratch due to its model size and sufficient amount of public text data [2]. However, this approach presents its own challenges. Firstly, LLMs are characterized by their substantial model size, and federated learning inherently requires a longer duration when compared to centralized training methods [3, 4]. Secondly, given that many contemporary LLMs are generative models, concerns regarding privacy violations due to potential semantic information leakage existed even before the advent of federated learning [5–7]. This raises questions regarding the suitability of federated learning for developing generative LLMs.

An alternative to fine-tuning is the Retrieval Augmented Generation (RAG) approach [8], which customizes an LLM for specific datasets or tasks. This technique begins by identifying the document that best matches an incoming query.

Then, it integrates this query-document pair into the model to generate responses for questions that the LLM may not have prior knowledge of. This method does not involve additional training, ensuring the model cannot retain semantic information. Furthermore, it offers a privacy advantage, as documents for searching can be pre-filtered based on user access rights.

Implementing RAG-based LLMs in a federated environment raises questions about their effectiveness in providing answers. A key challenge is that each client’s answer is restricted to its data. For example, in a collaborative hospital scenario, if asked, “*Which hospital has the most patients?*” individual hospitals cannot provide accurate answers as they only possess their data. To address this limitation, we introduce FedRAG, a RAG-based LLM that leverages Global-Local Question-Answering techniques to enable automated collaboration in a federated environment.

2 FedRAG

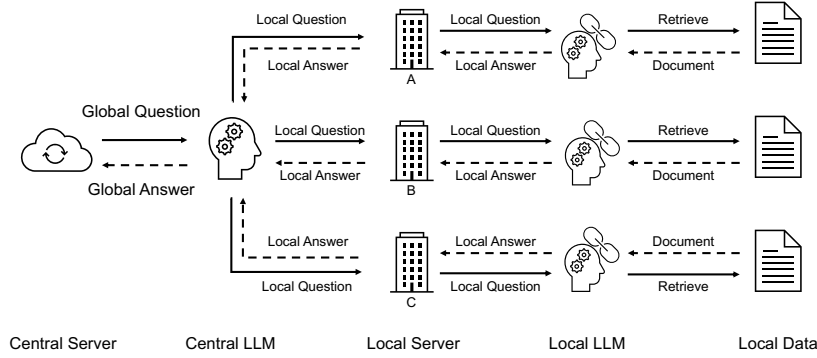


Fig. 1. Overview of Question Answering Process in FedRAG

2.1 Workflow

To comprehend the concept of FedRAG, it is essential to delve into the fundamental training process of federated learning. In a conventional server-client configuration, the central server disseminates a global model to clients, who employ their data for local training. Following this local training, the clients return their local models to the server, which aggregates them, typically using a method like FedAvg [9] to update the global model.

FedRAG adopts a similar approach but concentrates on exchanging questions and answers rather than models, as illustrated in Figure 1. Initially, a global question is dispatched to the central server. Utilizing the LLM on the

central server, this global question transforms into a local question, with the specifics of this transformation to be elucidated later. Subsequently, each client generates a local answer using the RAG approach, employing the modified local question. These local answers are then transmitted back to the server. The LLM on the central server synthesizes a global answer, harnessing both the local answers provided by clients and the initial global question, thereby ensuring a comprehensive and well-rounded response.

2.2 Global-Local Question-Answering

Returning to the earlier hospital scenario, “*Which hospital has the most patients?*” can be considered a global question. When it transforms a local question, it becomes, “*How many patients are in your hospital?*”. Then, each client will provide an answer based on the specific number of people (local answers) in their respective hospitals. The central server will then aggregate these numbers to determine the hospital with the most patients, arriving at the final answer through its LLM.

We propose the following prompt engineering approach to automate the transformation of a global question into a local question. Essentially, we define the desired global-to-local relationship based on one example. Subsequently, we present the global question we intend to transform, enabling the LLM on the central server to comprehend the intent and generate the corresponding local question.

Prompt: Question Transformation

Transform the question like the following relationship.

Global question: Local question =

“Where is the biggest country?” : “What is the size of your country?”

Global question: [INSERTED GLOBAL QUESTION]

Table 1 displays the global questions transformed using the aforementioned prompt engineering technique.

2.3 Implementation of FedRAG

We implemented FedRAG using Streamlit (GUI), Flask (server), OpenAI GPT3.5 (LLM), and LangChain (LLM framework). To make our demonstration accessible to anyone without requiring specialized knowledge, we selected Wikipedia pages from three countries and saved them in PDF format. These PDF documents were then configured for retrieval as part of our demonstration. In this setup, three local servers (clients) are each responsible for managing their respective datasets. Additionally, a global server serves as an interface to the streamlit application. Ideally, each server and client should host its own LLM independently to minimize exposure to third-party interactions. However, for simplicity

Table 1. List of Global and Local Questions Transformed

Global Question	Local Question
What novel technologies are transforming patient care?	Has your clinic implemented any new technologies to enhance patient experience recently?
How are nations addressing cybersecurity threats?	What cybersecurity measures has your local government office adopted?
What is the impact of cryptocurrency on traditional banking worldwide?	Does your bank offer any services related to cryptocurrency trading or storage?
How are tech companies addressing issues related to data privacy?	What data protection features does your company’s product offer to users?
What methods are being globally recognized to improve online learning?	How has your school adapted its teaching methods to facilitate online learning?
What global trends are emerging in e-commerce?	Has your retail store enhanced its online shopping platform recently?

of implementation, we have chosen to utilize OpenAI’s API. The implementation, including detailed source code and a process recording, can be found in our GitHub repository¹.

3 Conclusion and Future Work

In this research, we embarked on the development of a Federated Large Language Model system using the RAG architecture, all without requiring extra training steps. In a world where data privacy is increasingly significant, and as LLM performance continues to advance, we acknowledge the demand for a more secure and streamlined approach to LLM implementation. To tackle this challenge, we introduced FedRAG and harnessed a Global-Local question-answering approach. Through this method, we amalgamated responses from decentralized clients to produce a global answer that closely resembles the result obtained from scanning the entire dataset.

Our directed research encompasses two distinct domains. The first area involves substantiating our claims through systematic experimentation. In this study, our primary objective was to clarify our point with an intuitively comprehensible example. Nevertheless, further experiments are required, encompassing diverse question formats and comparing a single model with unrestricted access to all data against a scenario where access is limited through FedRAG.

The second domain concentrates on enhancing privacy safeguards. While we previously highlighted the importance of categorizing accessible documents for individual clients, we propose elevating this process to a new level of transparency by incorporating technologies like blockchain and smart contracts. Additionally, instead of merely regulating access rights at the document level, we anticipate achieving more advanced privacy protection by granting access rights at the contextual or word level, achieved through additional prompt engineering.

¹ <https://github.com/thejungwon/FedRAG>

References

1. S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, “Sparks of artificial general intelligence: Early experiments with gpt-4,” *arXiv preprint arXiv:2303.12712*, 2023.
2. W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, “Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning,” *arXiv preprint arXiv:2309.00363*, 2023.
3. H. T. Nguyen, V. Schwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, “Fast-convergent federated learning,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2020.
4. J. Jiang, X. Liu, and C. Fan, “Low-parameter federated learning with large language models,” *arXiv preprint arXiv:2307.13896*, 2023.
5. S. Kim, S. Yun, H. Lee, M. Gubri, S. Yoon, and S. J. Oh, “Propile: Probing privacy leakage in large language models,” *arXiv preprint arXiv:2307.01881*, 2023.
6. X. Yuan, X. Ma, L. Zhang, Y. Fang, and D. Wu, “Beyond class-level privacy leakage: Breaking record-level privacy in federated learning,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2555–2565, 2021.
7. L. Zhang, B. Shen, A. Barnawi, S. Xi, N. Kumar, and Y. Wu, “FeddpGAN: federated differentially private generative adversarial networks framework for the detection of covid-19 pneumonia,” *Information Systems Frontiers*, vol. 23, no. 6, pp. 1403–1415, 2021.
8. K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, “Retrieval augmented language model pre-training,” in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.
9. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.