Master Thesis

Cyber Security & AI

Radboud University

Federated Learning for Secure and Permission-Aware LLM Applications

Author:
Pien Rooijendijk
s1054190
pien.rooijendijk@ru.nl

Supervisor:
Dr. Stjepan Picek
stjepan.picek@ru.nl

External Supervisor:
Dyon van der Ende
dyon.vanderende@foxcrypto.com

Second assessor:
Dr. Güneş Acar
g.acar@cs.ru.nl

September 3, 2025

Contents

List of Figures

List of Tables

List of Abbreviations

LLM	Large Language Model
NLP	Natural Language Processing
FL	Federated Learning
RAG	Retrieval Augmented Generation
PEFT	Paremeter-Efficicient Fine-Tuning
LoRA	Low-Rank Adaptation
DP	Differential Privacy
CDP	Central Differential Privacy
LDP	Local Differential Privacy
HE	Homomorphic Encryption
SA	Secure Aggregation
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role-Based Access Control
ABAC	Attribute-Based Access Control
PBAC	olicy-Based Access Control
FedIT	Federated Instruction Tuning
RL	Reinforcement Learning
SEAL	Simple Encrypted Arithmetic Library
TenSEAL	Tensor Simple Encrypted Arithmetic Library
GDPR	General Data Protection Regulation
PBAC	Policy/Attribute-Based Access Control
IND-CPA	Indistinguishability under Chosen Plaintext Attack
CKKS	Cheon-Kim-Kim-Song Encryption
FAISS	Facebook AI Similarity Search
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
BLEU	Bilingual Evaluation Understudy

Abstract

This thesis researches the integration of Federated Learning with Large Language Models, with the aim of enabling privacy-preserving training and inference on sensitive, distributed data. A practical framework is developed that combines Parameter-Efficient Fine-Tuning using Low-Rank Adaptation, retrieval augmentation methods, and security mitigations such as Homomorphic Encryption and Differential Privacy. This setup ensures that sensitive documents never leave the client environment, while encrypted model updates are securely aggregated at a central server.

Furthermore, the proposed system leverages permission-aware retrieval, enforcing document-level access control during inference. This prevents the exposure of unauthorized data while still providing meaningful local reasoning and collaboration across clients. Evaluation is conducted using both lexical metrics (BLEU and ROUGE) and retrieval-specific scores (IXN), as well as semantic measures (RAGAS), to assess answer quality and security enforcement. Experimental results show that while access control may reduce answer completeness and accuracy in some cases, the framework effectively prevents information leakage and maintains accurate response quality for authorized users.

Acknowledgements

Starting this thesis, I was not sure if I could finish it before summer. I had set ambitious goals: I wanted to create something practical for my internship company while also making a meaningful contribution to research. After scaling down the project, some peers still told me I was asking a lot of myself. Some even joked that I had already started my PhD with this large topic. There is no "real" end to this thesis, but I hope it serves as a stepping stone towards what should become standard in this era of using LLMs.

I would like to express my gratitude to everyone who has supported me throughout the completion of this thesis. First and foremost, I thank my supervisor, Stjepan Picek, for the guidance, constructive feedback, and discussions. I am also grateful to my colleagues and peers at the company where I did my internship for the discussions, collaborative spirit, and moral support throughout this journey. I am particularly grateful to my external supervisor Dyon van der Ende for the thoughtful suggestions, valuable guidance, and excellent suggestions.

Moreover, I am especially grateful to Martin Hensbergen, who became my unofficial third, or maybe fourth, supervisor, for discussing ideas, pushing me to think critically, and helping me consider edge cases. A big thank you to Sander Dorigo for providing me with this project and giving me the freedom to make it my own. Your guidance and enthusiasm were what kept me motivated throughout this process.

Finally, I am deeply thankful to my family and friends for their support during my studies at Radboud. Special thanks go to my boyfriend, who thought alongside me, offered fresh perspectives, and helped me turn this thesis into a coherent whole. Everyone kept me going, and this thesis would not have been possible without them.

Chapter 1

Introduction

What if it were possible to summarize hundreds of pages of classified documents in a matter of seconds, without ever revealing the content to users without access? Artificial Intelligence (AI) systems are advancing to perform tasks like these without requiring human intelligence [1]. AI has its roots in 1940 with early theories on computation using processes similar to neurons in the human brain [2]. Over the decades, various types of AI, machine learning (ML) in particular, have significantly transformed the field in sectors like healthcare and cybersecurity.

Large Language Models (LLMs) have become experts in various applications, ranging from natural language processing (NLP) to customer service automation and generation tasks. However, using them also brings security risks, such as data leaks, adversarial attacks, and unauthorized access. When organizations handle highly classified information, including state secrets, this data must be stored and processed safely [3]. One mitigation to ensure maximum security is to handle the data in an air-gapped network [4]. These networks contain numerous classified documents, and some of them are many pages long. A significant amount of time could be saved if a reliable LLM could generate accurate summaries of these documents without granting other users access to the original content when requesting the same summary. If a user of an LLM application were able to reproduce classified information despite only having the clearance of a normal user, it would pose a serious risk. The problem becomes complex given that these systems do not have a built-in mechanism to adhere to permission constraints. Some systems use prompt instructions to restrict LLM behavior explicitly, such as *"Do not answer questions about unauthorized documents."*. But depending only on prompt-based rules is unreliable, as LLMs may fail to consistently adhere to these constraints. A better solution is to use a combination of methods, such as filtering data and applying permission controls across multiple layers, and aggregating the results for the LLM to process.

Training and fine-tuning LLMs to be used on classified documents requires many computational resources. Models like GPT-4 and Gemini cost tens of millions of dollars to train, in terms of hardware, energy, and staff expenses [5]. Even when using open-source LLMs such as Llama, these constraints remain, setting a boundary on the possibility for smaller institutions to perform full fine-tuning using LLMs due to these computational and financial constraints. The traditional manner of fine-tuning involves sending large language model parameters and datasets, which becomes impractical in these environments. This challenge arises particularly in centralized training, where limitations such as bandwidth and storage can create a bottleneck in the process. These issues can be resolved with Federated Learning (FL) by enabling decentralized model training. With FL, clients perform local training on private data and only share their model parameters with a central server. This approach reduces the transfer of the large parameters of LLMs and supports privacy goals by design [6].

However, there are still challenges with standard FL in terms of efficiency and security. In

decentralized environments, it is often intensive in terms of bandwidth to transmit the full model parameters for LLMs. Furthermore, sharing all updates leaves the client vulnerable to privacy attacks, such as gradient inversion attacks, which can use the shared updates to reconstruct the client's private data. To address communication bottlenecks, we introduce parameter-efficient fine-tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA). This approach lowers computation and communication costs by allowing users to modify only the trainable parameters rather than the full model [7, 8]. However, privacy risks remain, as inference attacks are still feasible even when LoRA is applied. Attackers can reconstruct sensitive training data from the shared LoRA gradients, since these still encode information about token representations and co-occurrence patterns [9]. As a result, additional privacy-preserving techniques for end-to-end security in a federated LLM training have been proposed. Differential Privacy (DP) adds noise to model updates, thereby limiting information leakage [10]. Homomorphic Encryption (HE) enables computation on encrypted parameters, providing strong confidentiality without introducing additional noise [11]. We suggest a new federated fine-tuning framework that combines FL, LoRA, DP, and HE for a secure and private training across private datasets containing classified information, building on these privacy-preserving techniques.

The primary problem statement of this master's thesis is to design an FL framework that strictly enforces permissions within an organization to prevent unauthorized access to sensitive data, particularly for state-secret-level documents. Additionally, LLMs should operate securely, within an air-gapped network, without compromising performance or security.

Research Questions

The following are the proposed research questions that will be answered in this thesis:

1. How can federated learning be used to train LLMs on sensitive data without exposing that data?
   (a) How can FL be adapted to support PEFT methods, in particular LoRA for LLM fine-tuning across clients?
   (b) What mechanisms during training ensure that the client data is still protected, while still contributing to the global model improvements?
2. What are the risks of using LLMs in a federated learning setup without taking security into account, and what mitigation can be used?
   (a) What types of adversarial threats are possible in this FL setting?
   (b) What mitigations can be used to defend against these threats?
3. How does permission enforcement affect the accuracy of LLM responses for authorized clients?
   (a) How effective is permission enforcement in ensuring that the clients only receive answers that are based on their private documents?
   (b) What is the impact of restricted context windows on the response accuracy for different question types?

The following main contributions were made by answering the above research questions. We propose a secure federated LLM framework that integrates HE with LoRA adapters. It addresses security vulnerabilities during training, under the assumption of an honest-but-curious server. The key features include:

1. **Parameter-efficient updates**: Only LoRA adapter layers are fine-tuned and shared, minimizing both computational overhead and communication costs, making the framework suitable for environments with resource limitations.

2. **Limiting data leakage**: DP is added to the LoRA gradients before encryption, offering a privacy guarantee and limiting the potential for information leakage from the individual client updates.

3. **Strong data privacy through HE**: By encrypting the model updates through HE, clients were ensured that only encrypted information reaches the server. This offers end-to-end confidentiality without solely relying on DP.

A comprehensive evaluation is done on the federated `DeepSeek-R1-Distill-Qwen-1.5B` using evaluation metrics for permission enforcement and response quality. Our results demonstrate that the proposed framework effectively enforces access control during retrieval, significantly limiting unauthorized access to sensitive documents while maintaining an acceptable response quality for users with the correct permissions, as measured by BLEU, ROUGE, and RAGAS scores. The code, data and results are available in a GitHub repository [1].

The remainder of this thesis is structured as follows: Chapter 2 introduces a comprehensive background to provide the knowledge to understand the subsequent material. Chapter 3 provides a comprehensive review of the related work. Chapter 4 presents the proposed Secure Federated LLM framework. Chapter 5 presents the chosen model, dataset, data pre-processing, and privacy/security implementation details, followed by Chapter 6, which contains the results. Chapter 7 offers future work and a discussion, also including limitations and implications. Finally, chapter 8 will conclude this research.

[1] `https://github.com/pcrooijendijk/Master_Thesis`

Chapter 2

Background

In this thesis, an end-to-end secure federated large language model (LLM) framework is designed to operate under strict privacy constraints. To better understand how this framework is built, the required knowledge is described in the next section. First, federated learning (FL) will be addressed as the core of the training mechanism, which enables decentralized optimization across clients without exposing their sensitive data. Building on this, permission control will be enforced to ensure that only authorized users can access the sensitive data. Then, to enable question-answering over sensitive data, retrieval-augmented generation (RAG) is integrated into the pipeline. To optimize efficiency for computation and communication overhead, parameter-efficient fine-tuning (PEFT) using LoRA will be applied. Given rising security threats, ranging from gradient leakage and data poisoning, the framework also incorporates the following defenses: differential privacy, secure aggregation via homomorphic encryption, and access enforcement.

2.1 Federated Learning

The term FL was first mentioned by McMahan et al. [6], who cite that it is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem under the coordination of a central server or service provider. Each client's data is stored locally and is not exchanged or transferred with the other clients. Instead, updates are intended for the servers' aggregation to achieve the machine learning objective. This process of aggregating involves combining the local models from the clients into a single global model, while maintaining the clients' privacy [12]. In contrast to the traditional privacy protection algorithms, DP, and k-order anonymity, FL offers a training method that uses clients to secure the privacy of its users [13]. Instead of relying on a third party to store data, FL primarily preserves the privacy of its users by exchanging the weights of their model using encryption methods [14]. Centralized approaches pose challenges when the data has a sensitive nature [15].

When combined with LLMs, FL helps conserve both computational and data resources. This eases the workload for each client during training and inference and boosts the language models' ability to perform different tasks [16]. When these models are used for sensitive data, they cannot store classified or private data locally in a centralized manner for training due to privacy reasons. FL solves this challenge by keeping the data local for each client while still enabling improvements by using a global model. This is critical given the known security risks of LLMs, including prompt injections, model stealing, or gradient leakage, making decentralized training essential for safe deployment of LLMs using private data.

2.2 Large Language Models

LLMs such as the Generative Pre-trained Transformer (GPT) and the Llama series [17, 18], form the foundation of many modern language understanding and generation systems. These models can generate human-like language based on given context since they are trained on massive amounts of text [19]. While LLMs show impressive performances on language-based tasks, they often lack the up-to-date knowledge and access to real-time information, which may lead to difficulties with certain tasks. These models are slow to react to updated information, and it is computationally heavy to fine-tune an LLM on every dataset update. Another downside of such models is that they are prone to hallucinations: the model produces outputs which may seem correct and plausible but are in fact either not the user's intent or are factually incorrect [20]. Pre-trained language models tend to produce hallucinations when they do not have access to an external memory, solely based on the parametrized implicit knowledge base.

LLMs inherently have two major drawbacks, which are hallucinations and their hardware memory size. Hallucinations in the models originate primarily from a lack of knowledge; the models generate information based on their intrinsic knowledge, which is their trained dataset. Ideally, the models should generate from knowledge or a source, or decline to answer the query rather than hallucinating their response. Ultimately, the answer should be reliable, and information must be correctly sourced or learned to give a proper response to the query of a user, which in turn is ready for usage in decision making, for instance.

The other disadvantage of using LLMs in this context is their size. DeepSeek, which this research uses, with 1.5 billion parameters, is too large to be trained end-to-end in a federated learning setup on conventional hardware. To mitigate this constraint of limited resources, there needs to be a method to minimize the amount of parameters that are sent back and forth, where PEFT can be used (see section 2.4).

2.3 Retrieval Augmented Generation

To prevent the aforementioned problem of hallucinating, RAG was introduced by Lewis et al. [21] in 2020, enabling models to generate responses grounded in external information sources, to ensure extra knowledge on what the LLMs already know without additional training.

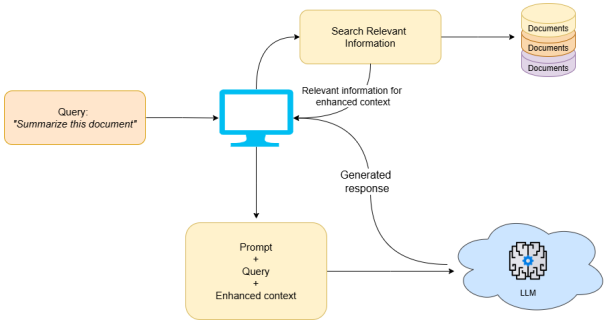

Figure 2.1: This high-level overview of RAG consists of two primary stages: retrieval and generation. In the retrieval stage, a query from the user is used to search an external document index using dense or sparse vector similarity. The top-k most relevant documents are retrieved and concatenated with the original prompt to form an augmented input. In the generation stage, this augmented input is processed by the LLM to generate a response using this external context.

A diagram of RAG can be seen in Figure 2.1, which shows the main concept. To begin the setup, a retriever module is used to process the input query and search a vector that is built from external documents. The retriever selects the top-k most relevant documents based on the similarity score. The original prompt is then extended with the retrieved documents and is then passed to the LLM. This enables it to generate responses based on both internal and external parameters retrieved in real time.

Using RAG can thus be beneficial to circumvent hallucinations and ensure that the LLM has the most up-to-date knowledge to answer the query without requiring retraining the entire model or fine-tuning it.

2.4 Parameter-Efficient Fine-Tuning

One of the drawbacks of FL for LLMs is the massive amounts of weights, which result in high computation and communication costs, making classical FL impractical [22]. According to Kuang et al., a full-parameter fine-tuning LlaMA-7B in FL requires 28 Gigabytes of message transfer for one round of communication between the clients and the server [23]. PEFT enables efficient adaptation of LLMs to downstream tasks without the need to use and fine-tune all the parameters of LLMs. The majority of the pre-training parameters are fixed, resulting in a significant reduction of computation and storage cost [24]. This method only fine-tunes a small subset of additional parameters of the model. Another similar method to this is Low-Rank Adaptation (LoRA), which freezes the original model parameters, and only trains the parameters of the new added layers to achieve similar fine-tuning results [8, 25]. LoRA is specifically efficient when the model has been trained on a large dataset and must be fine-tuned on a smaller dataset for a specific task [15].

2.5 Gradient Leakage

FL protects the security of the data, which can lead to a secure transmission of models while protecting the local sensitive data. However, there is still one prominent type of security risks against an FL architecture that threatens the privacy of the data of clients, namely gradient leakage.

For a long time, it was believed that gradients are safe to be shared, and the training data could not be leaked when exchanging the parameters. Zhu et al. [26] demonstrated that the private training data can be obtained from the shared gradients. The greatest risk here is training with a centralized server where the data is located in one place. Language models process the generated text in embeddings, when employing LLMs and sharing their raw gradients between servers, original words from the training data can be uncovered by predicting the original value of the words from the given context where in each sequence 15% of the words are replaced with a mask token [26]. The classic variants of reconstructing the gradients are either optimization based on reconstruction or a reconstruction where the attack reconstructs the input from the parameters using knowledge about the structure of the model [27].

2.6 Federated Learning Security Defenses

The following sections describe the three security defenses to mitigate the risks of poisoning and gradient leakage. Together, these methods can form a framework for a secure and robust defense for LLMs in adversarial setups.

2.6.1 Differential Privacy

Since there is a possibility of recovering the original data from the parameters of the models, these weights should be protected from possible eavesdroppers by using differential privacy (DP). This method ensures that every single document is protected by adding perturbed noise to either the data or the weights while ensuring that complex computations over the dataset are still feasible [28]. Perturbing the data means that the dataset can still be analyzed and even compute the median, mode, and mean over it, which does not reveal any information about the individual data points.

Expressed in formulas having DP means that a randomized mechanism $M : \mathcal{D} \to \mathcal{R}$ is essentially (ϵ, δ)-differentially private for $\epsilon > 0$ and $\delta \in [0, 1)$ if for any two dataset which differ at most one sample $D, D' \in \mathcal{D}$ and for any subset of outputs R $\subseteq \mathcal{R}$ it holds that [29]:

$$\mathbb{P}(M(D) \in R) \leq \exp(\epsilon)\mathbb{P}(M(D') \in R) + \delta \tag{2.1}$$

Protecting the privacy of the architecture using DP can be done in three different ways: creating a privacy-preserving document dataset, protecting the client's privacy during the training, or producing models that are resistant to model inference and model inversion attacks [10]. DP itself can then also be divided into two categories based on the location where the noise is added: central differential privacy (CDP) [30] and local differential privacy (LDP) [31]. CDP ensures that there is noise added on the server side, and LDP adds its noise to the client side before sending the updates [32]. When comparing LDP with CDP, CDP is not sufficient to protect the participant training data from gradient leakage at document-level [33]. LDP is a stronger privacy guarantee because it does not require the clients to trust the aggregator the correctly apply noise. However, LDP requires more noise to be added to all clients compared to adding the noise at the server side.

The risk of privacy disclosure caused by adding a single document is controlled in a small and secure range where the attacker cannot obtain accurate information about the data during the training process, minimizing the risk of a poisoning attack [13]. This mitigation does not entirely prevent information leakage when the model's parameters are known, which suggests a need for cryptographic protections.

2.6.2 Homomorphic Encryption

Homomorphic encryption (HE) focuses on encrypting the data with the capability for computing over this data without having access to the secret key [13]. The operations on the ciphertexts are equal to the plaintext operation results. Before the training of the clients begins, an HE key-pair is synchronized across all clients through a secure channel [34]. Each client encrypts its parameter updates using the key and uploads the resulting ciphertext to the server. The server aggregates the client updates by leveraging the addition property of HE and sends back the result to all clients. Since each client also holds the corresponding private key, they can decrypt the received updates and update the local models accordingly. However, HE performs complex cryptographic operations which are extremely computationally heavy [35]. According to Zhang et al., 80% of the training iteration time in FL using HE is spent on encryption and decryption [11]. They propose to mitigate this limitation by using batches of parameters instead of individual weight encryption. The training process is accelerated while the accuracy loss is minimal.

2.6.3 Secure Aggregation

Secure Aggregation (SA) is a scheme that defends the global model in FL against inversion or inference attacks, where the server is only allowed to learn the average of the updates of the clients

[36]. In general, SA generates a mask for the local models using random keys before aggregation [37]. By using secret sharing, the server reconstructs the private seed of each collected client and computes the aggregated model [38]. Both DP and HE can be used for securely aggregating. In DP, noise is added to their local models, while in HE, client encryption is added to the updates [39].

2.7 Access Control and Permissions

Access control policies, that specify which user has access to which information, must be defined in every database within organizations [40]. Several access control models have been developed over the years, each specific to different needs and security goals. These methods include Discretionary Access Control (DAC) [41], where data owners define the access policies for other users, and Mandatory Access Control (MAC) [42], which enforces access based on various classification or security levels like top secret, secret, confidential, and unclassified. Flexibility and security are compromised when using DAC and MAC when scaling these across diverse users in modern collaborative environments.

In contrast, Role-Based Access Control (RBAC) provides a balance between security and scalability. In RBAC, permissions are assigned to the users roles rather than to individual users [43]. Users can gain access by being assigned one or more roles. This ensures that users access only what is necessary for their roles, which makes this structure manageable in complex systems. RBAC assigns access rights to roles such as viewer, editor, or admin. A practical example of this RBAC enforcement can be found in Confluence, an enterprise collaboration platform. Confluence divides permission into global permissions, which apply across the entire page, and space permissions, which control access to specific content areas [44]. Each space can be managed independently which users or groups have access to view, edit, or administer its content and documents.

Chapter 3

Related Work

The intersection of FL, LLMs, and access control for sensitive data continues to be an emergent research area. While extensive literature already exists on each of these topics, their integration with high-security environments remains limited. The key advancements across these three domains is highlighted in this section.

3.1 Federated LLMs

There have already been some advances in the field of using FL for LLMs; the FATE-LLM architecture [45] represents a production-ready solution extending the FATE ecosystem. This framework supports both homogeneous and heterogeneous large language model federation, incorporating parameter-efficient optimization techniques such as LoRA and P-Tuning-v2 [45]. However, FATE-LLM does not implement access controls or security requirements for sensitive data. Although the framework is focused on the common security and privacy risks, it is left with the vulnerability to gradient leakage, where an inversion attack is possible, a critical gap for classified data.

Federated Instruction Tuning (FedIT) leverages FL to instruction-tune LLMs, improving their generalizability while maintaining data privacy [15]. Zhang et al.'s approach supports fine-tuning of client queries by using instruction-response modules. The evaluations are done by GPT-4 and show that their method exceeds centralized training when using limited local data. Zhang et al. [46] further refined FedIT, which has an auto-evaluation mechanism and is trained against data extraction attacks. Another federated LLM architecture was developed by Kuang et al. [23], which offers an end-to-end FL fine-tuning mechanism that also uses PEFT with extensive evaluation methods.

3.2 Federated Learning for Sensitive Data

Without compromising privacy or requiring centralized data storage, FL provides an approach for training models on sensitive data. This is especially crucial in the following domains where data confidentiality is crucial due to regulations such as the GDPR [47].

3.2.1 Healthcare

Extensive research has already been conducted on the application of federated learning to clinical data. Clinical-domain LLMs, such as ClinicalBERT, Med-PaLM, Almanac, and Meditron, are examples of these models that are essential for question-answering tasks within healthcare [48–52]. However, these

models often assume a central medical knowledge base, which is not always realistic in distributed healthcare settings [49, 53]. The viability of using FL in healthcare is seen in research like EXAM, a model trained across 20 international features to predict the requirement of oxygen for COVID patients, which achieved a high AUC score and improved the generalizability when compared to other centralized models [54]. Using sensitive patient data, other federated systems have been created to classify patients with melanoma or tuberculosis and predict ICU mortality [55, 56].

Teo et al. [57] conducted a review of the real-life applications of FL in the healthcare sector by studying over 600 research papers. The results revealed that most of the papers are still a proof-of-concept, but FL is already deployed across various domains such as radiology and internal medicine [57]. To address the challenge of using data fragmented from different health institutions, which are under strict privacy regulations, Jiang et al. [49] proposed MediRAG, which is distributed across multiple hospital servers. MediRAG uses authentication and policy-based access enforcement to ensure secure query routing. It also performs retrieval and vectorization on the patients' datasets, and only shares carefully filtered results and embeddings with a central LLM. This framework already takes the privacy requirements of the sensitive data of the patients into account.

By combining federated retrieval protocols with access control mechanisms and optional encryption, the clinical federated LLMs facilitate distributed question-answering without compromising the patients' privacy. This caused a shift in using centralized corpora, which was the basis of a fully regulated FL-RAG system. Such systems should be GDPR compliant before they can be used across hospitals for research without raw data sharing [47].

3.2.2 Industrial Engineering

Since data in the industrial engineering area is not available directly due to some constraints of laws and regulations, they often opt to use federated learning [58]. The data fragmentation here happens across supply chains, manufacturing ecosystems, and infrastructure networks. Hu et al. [59] developed Federated Region Learning, which integrates distributed sensor data from air or water monitors across different urban environments. When the models were trained locally on geographically partitioned data and then aggregated on region-specific features, the decentralized architecture achieved an increase of 23% in accuracy when predicting pollution compared to centralized alternatives.

3.3 Access Control in Federated Systems

Access control is an essential component of federated systems, as it determines how data and model updates are shared across participants without violating privacy or security policies. From the traditional Role-Based Access Control (RBAC) to the more adaptable Attribute-Based Access Control (ABAC) and Policy-Based Access Control (PBAC), existing research has examined a variety of approaches [60, 61]. PBAC allows policies to be formally defined and implemented, which has been proven useful in domains such as healthcare [53].

Particularly in federated learning, access control mechanisms are used to regulate permissions for sharing gradients and contributing model updates in addition to controlling data access. Prior work has highlighted the challenge of enforcing fine-grained control without compromising performance or usability. For example, healthcare-focused federated frameworks often restrict access to only subsets of Electronic Health Records, ensuring compliance with legal and ethical standards while still enabling federated and collaborative model training [62, 63].

Chapter 4

Methodology

The following chapter presents the methodology in which the approach is discussed that combines multiple techniques to support secure and permission-aware training and inference across distributed clients without exposing raw data. The system design uses PEFT, specifically LoRA, and combines this with privacy-preserving mechanisms such as DP and HE. The sections that follows, describes the threat model considered, the details of the proposed framework, privacy mitigations, and the implementation of the user interface.

4.1 Threat Model

The overall threat model consists of two parts: threat model during the *training phase* and the *inference phase*. In the context of the *training phase*, privacy refers to the presence of malicious clients and an adversary that seeks to learn the weights. The other threat model, *inference phase*, at the answering side, where we do not want the model to answer anything about other clients' documents for which they do not have permission. Not only is the overall threat model of this architecture considered, but also the general threat model of FL needs to be considered.

The server or aggregator is honest-but-curious, which is a common threat model used in the existing FL architecture [64, 65]. Adversaries attempt to extract private data files from other clients while following the protocol correctly. In this protocol, both the client's data and the cloud's model parameters are kept private under this threat model. The server is not required to be a trusted party, and the privacy of the data is preserved since the client's secret is not known. As the parameters are decrypted, the server cannot learn anything about the client's data. When the clients have malicious intent, they may deviate from honest clients by sending random updates, replay updates, or removing updates when communicating with the server [66]. Under this assumption of an honest majority, the parameters appear random after decryption. The encrypted parameters provide indistinguishability under a chosen plaintext attack (IND-CPA), meaning the server cannot learn anything from what it observes, thereby guaranteeing the privacy of the client's data and the global model [67].

Adversaries can attempt to obfuscate the process by attacking from within or from outside. Overall, insider attacks are stronger than outsider attacks [68]. Insider threats are difficult to discriminate from benign activity within the organization or architecture [69]. In the proposed architecture (see Figure 4.2), insider threats can take the form of a malicious server or client. Outsider attacks may include eavesdroppers on the communication channels who use the weights to infer information about the training data. Zhu et al. [26] have already demonstrated that training data can be reconstructed from the weights for image classification and language modeling tasks, thereby proving that sharing parameters can leak private training data. Another threat to the FL architecture is poisoning attacks,

where we divide this attack into random and targeted attacks, where random poisoning attacks have the goal of reducing the accuracy of the FL model, and the targeted poisoning attacks aim to have the FL model classify the target label set by the adversary [70].

The primary potential threats are outlined below and illustrated in Figure 4.1, each labeled accordingly. These represent the most commonly cited threats in the literature [26, 27, 70]. While additional threats exist, their inclusion falls outside the scope of this thesis.

- **External Eavesdropper**: An adversary can eavesdrop on the weights by using the communication channels of the clients and the server. These weights can be used to extract information about the documents. An adversary can reconstruct the original training data and thereby the classified documents by using the weights.
- **Malicious Client**: Malicious clients can interfere with the benign model updates by poisoning the dataset, replaying the model updates, or removing updates. The ultimate goal of these clients is to corrupt the global model, with the consequence that the model's accuracy degrades, affecting the server's availability and integrity.
- **Curious Server**: A curious server may reveal training samples of a specific target when it receives the model updates, while it pretends to be a benign server. Furthermore, there is a possibility that the server can recover the original training data without using prior knowledge of these files [71]. Recovering the data files requires a corresponding decoder that reveals the pattern of the encoded patterns in the model's parameters.

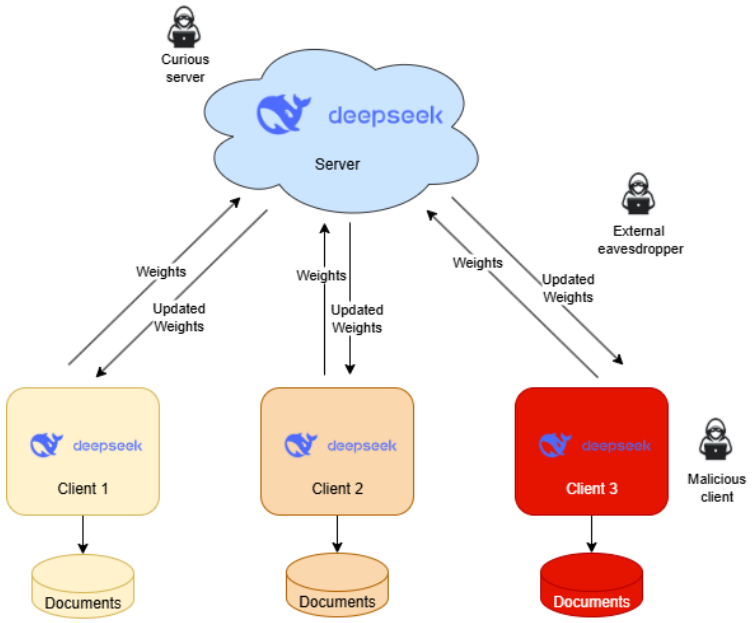

Figure 4.1: Threat model illustrating both internal and external threats, including a curious central server, an external eavesdropper, and a malicious client.

The threat model of using FL for LLMs in the framework, as mentioned above, is summarized as follows:

- **Adversary's Capabilities:** We assume that an adversary can eavesdrop on the weights by sniffing the network, making them an external eavesdropper. Additionally, a malicious client can interfere with the updates of the model by poisoning the dataset, thereby affecting the global model. Furthermore, a curious server can also disclose the parameters holding the gradients from the training data of the clients.
- **Adversary's Knowledge:** The adversary has no prior knowledge of the dataset or the model's parameters, nor does the adversary know the model's architecture, parameters, or training algorithm. At inference time, the adversary is allowed to query the model, as any client, to exploit possible backdoors.
- **Adversary's Goal:** The goal of an adversary in this context is to obtain classified data by using their knowledge of the system and their multiple capabilities in the training and inference phase to exploit the system. During training, the adversary can exploit the system by posing as the external eavesdropper, malicious client or curious server, leading to either gradient leakage or poisoning the global model.

4.2 Proposed Architecture

The global federated architecture is sketched in Figure 4.2, where no privacy or security aspects were taken into account. As each FL scenario starts, the weights from the clients are sent to the server for aggregation. After aggregation, the updates are sent back to the users, where the weights are loaded into their local model. In most cases, users have one or more roles. The allocation to these roles can be stored in either Confluence-like roles or directory services, such as Active Directory. Users have access to various spaces, each containing documents. Each user can have multiple roles and access numerous spaces. The security or access levels should be accessible from the programming interface.

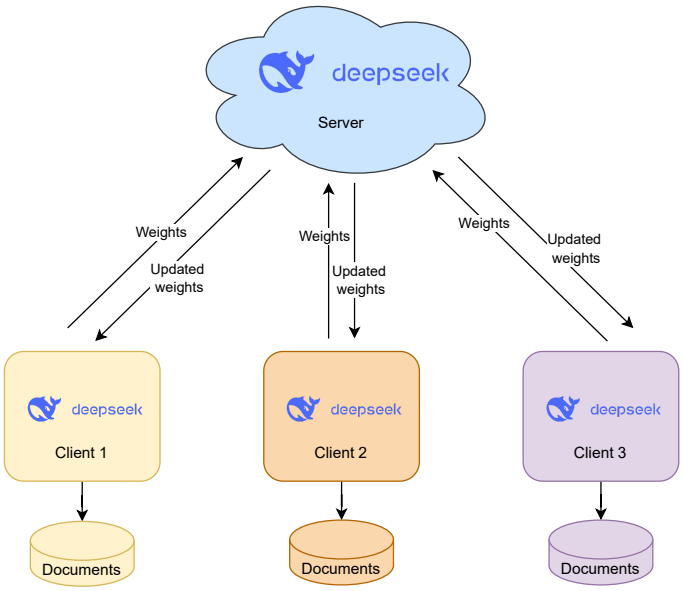

Figure 4.2: The proposed architecture with its clients having access to their local document store and a server which aggregates the weights and sends back the updates. Both the local and global models use DeepSeek.

4.2.1 DeepSeek

DeepSeek was selected as both the global and local models. In areas like writing, factual QA, and self-cognition, DeepSeek-R1 uses a multi-stage training pipeline and a small amount of cold-start data before fine-tuning on supervised data. Its ability to perform iterative token-based reasoning enables higher accuracy, particularly in tasks that require step-by-step analysis [72]. Setting the temperature between the optimal values of 0.6 and 0.8, DeepSeek-R1 is proficient in a various tasks, such as creative writing, general question answering, editing, summarizing, and more. Also, it demonstrates outstanding performance on tasks requiring long-context understanding, substantially outperforming DeepSeek-V3 on long-context benchmarks [73]. The model is encouraged through reinforcement learning (RL) to put its thinking process between "think" tags. According to the procedure of RL, a reward according to accuracy is given to the outcome [74]. The model aims to achieve robust reasoning capabilities without requiring any supervised fine-tuning data. This is a noteworthy achievement, as it underscores the model's ability to learn and generalize effectively through RL alone. DeepSeek allocates more thinking time to a problem by reevaluating its initial approach.

4.3 Baseline Federated LLM

In Figure 4.3, the baseline federated LLM framework is sketched. It serves as an insecure baseline where the user can query about the documents without taking any permissions into account. The documents still have confidential classifications, but the user can access them using RAG. The RAG

retriever also performs in a sense FL after retrieving the top-k documents. These relevant documents are used to extend the prompt to generate the LLM's response.

This framework serves as the baseline for enhancing the security of FL on LLMs. Modifying this insecure framework with security and permission control measures will enable us to implement the final secure architecture. The following describes a high-level overview of the baseline architecture:

1. The user queries the system, which is related to the classified documents. The client is already logged into the system; however, there are no additional authentication mechanisms in place to verify the user's identity.
2. The query is passed from the interface to the embeddings of the spaces. The query is passed to all spaces, including those with classified documents, because there are no security authorization mechanisms in place. The query will reach all decentralized spaces. There is no additional federation within the spaces. There are no security authorization mechanisms, so the query will reach all spaces regardless of user role attributes.
3. The top-k-relevant documents to the query are retrieved by embedding similarity across all documents within the spaces and are aggregated back to the RAG. There is no encryption or access control, so documents can leak or expose confidential information to unauthorized clients.
4. The documents are appended as context to the client's query and sent to the LLM for answer generation.
5. In the LLM's generation step, the LLM constructs a response from the query and the context, also showing the source document for enhancing interpretability.

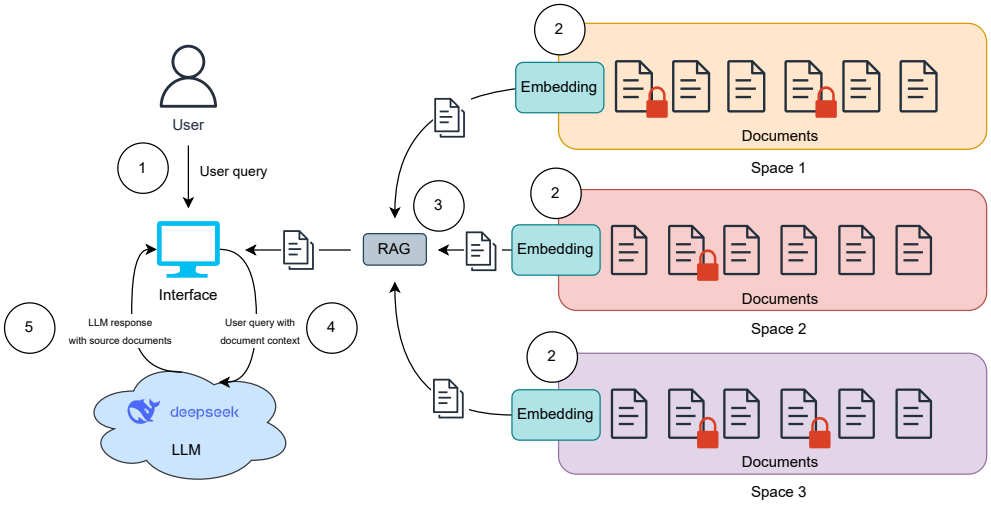

Figure 4.3: Baseline federated system architecture where the system allows the user to query across all spaces, including those containing classified data, without any authentication or authorization controls. Queries are processed without security controls, and relevant documents are retrieved based on their similarity to the query. As a result of this shortcoming in security measures, classified information may be exposed to unauthorized users. The retrieved documents serve as context for the LLM to generate responses, which also return the source documents.

4.4 Secure Federated LLM

Building on the baseline architecture described in section 4.3, the following section sketches the secure federated LLM framework that integrates user permission enforcement and FL-related security mitigations. The user is already assumed to be authenticated and has specific Confluence-like permissions (see Table 5.2). Access control is implemented to ensure control of user attribute-based access to classified documents during the retrieval of the decentralized spaces.

The following description is similar to the baseline, but in **bold** are the added security measurements to the process:

1. The user queries the system, which is related to the classified documents. The client is already logged into the system **and the user's identity is authenticated when logging into Confluence**.
2. The query is passed from the interface to the embeddings of the spaces. The query is passed to **only the spaces the user has access to**. The query will reach **the decentralized spaces which the user has access to. The query will only reach the spaces based on the role attributes of the client**.
3. The top-k-relevant documents to the query are retrieved by embedding similarity across all documents **the user has access to** within the spaces and are aggregated back to the RAG. **There is HE and access control, so no documents can leak or expose confidential information to the client, which is not authorized to view/access this data**.
4. The documents are appended as context to the client's query and sent to the **federated LLM, which is instruction-tuned,** for answer generation.
5. In the LLM's generation step, the LLM constructs a response from the query and the context, also showing the source document for enhancing interpretability.

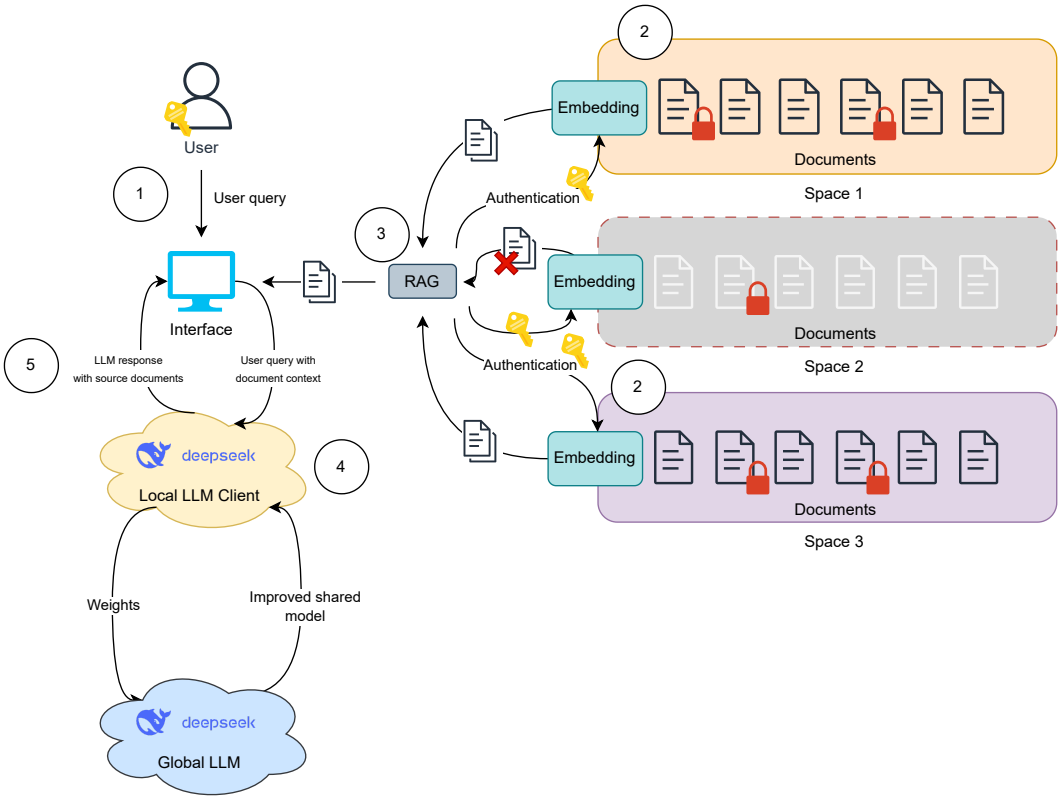

Figure 4.4: Secure federated system architecture where the system allows the user to query across spaces to which they have access. Queries are processed with security controls, and relevant documents are retrieved based on their similarity to the query. The retrieved documents serve as context for the LLM to generate responses, which also return the source documents.

4.4.1 Framework Architecture

The original framework is maintained for the proposed secure federated LLM architecture, as it serves as a building block that requires some additions to effectively become a secure federated LLM. Before querying, the clients are already authenticated when they get access to Confluence. Inherently, they have their own permissions, access to specific spaces, and, optionally, permissions based on their roles.

To address the challenges related to data privacy, computational overhead, and scalability, Secure Federated LLM integrates HE for secure computations and DP for adding noise to the gradients (see subsection 4.4.2 for the implementation). Furthermore, LoRA is implemented to reduce the number of trainable parameters when fine-tuning the LLM. FedIT[1] was used to improve the instruction-following capabilities of the secure LLM. FedIT already enables clients to fine-tune their local model using instruction-response pairs from their private datasets. This ensures that the model learns to generate responses that align with the user's custom instructions. Additionally, this approach maintains privacy by decentralizing the training data and securing the client's data.

[1]The code is available at `https://github.com/JayZhang42/FederatedGPT-Shepherd` [15]

As described in the overview in section 4.3, the server initially distributes the global model to the clients, where each client loads the weights into their local model and fine-tunes it on their private dataset using LoRA. During the training phase, the weights are subjected to additive noise, which prevents malicious clients from poisoning the weights. Then, to ensure secure communication, the client encrypts the parameters using the CKKS encryption scheme by TenSEAL. This encryption is done by encrypting their weights (w_i) by using the generated encryption context C:

$$c_i = \text{Enc}(w_i, C) \quad \text{for i} \in N \tag{4.1}$$

Using this encryption method ensures a safe transfer between the client and server. Additionally, the weights are safeguarded during the aggregation process at the server side without decrypting the gradients, as HE enables the performance of computations on the vectors. The parameters are aggregated as:

$$c_{\text{agg}} = \sum_{i=1}^{N} \frac{1}{n_i} c_i \quad \text{for i} \in N \tag{4.2}$$

where c_{agg} are the aggregated encrypted weights, n_i the number participating clients, and c_i the encrypted weights computed in Equation 4.4.1.

4.4.2 Privacy Mitigations

To enable security in the proposed framework, the following methods are implemented to mitigate potential threats as outlined in section 4.1.

Homomorphic Encryption

In this framework, HE is used to encrypt LoRA parameter updates before they are transmitted to the server. However, employing HE with tensors presents two challenges: efficiently encoding tensors before encryption and limited support for complex operations after encryption. Microsoft SEAL has an answer for supporting encrypted addition and multiplication, but is does not handle more advanced operations. A tensor can be encoded using a single ciphertext before encryption, ensuring it is optimized for both computability and memory usage. TenSEAL extends this functionality by providing the CKKSVector, which can hold $\mathbf{N}/2$ real values and supports addition and multiplication operationes allowed by SEAL. Additionally, TenSEAL supports power and square operations using an optimal circuit, thereby minimizing the depth required for multiplication.

Matrix operations, which are used in the FL training process, benefit from these optimizations. Dot products can be efficiently computed through multiple vector-to-matrix multiplications, which is based on the algorithm by Halevi & Shoup [75]. Together, these techniques enable privacy-preserving computation of LoRA updates while maintaining practical efficiency in a federated learning setting.

Differential Privacy

To also strengthen the privacy during the client's training, we incorporate DP, which modifies the gradients of the LoRA parameters at each training step at the client side. Adding the noise at the client side, makes it local DP (LDP). After the optimizer computes the gradients, using DP, the overall magnitude of the gradients is computed first. If this magnitude exceeds the preset threshold, the gradients are scaled down to ensure that they do not exceed the maximum norm; this limits the influence that a single data sample can have on the model updates (see Equation 2.6.1). This

clipping involves drawing noise from a Gaussian distribution, which is added to the gradients. This noise is scaled by a noise multiplier parameter, which controls the level of privacy protection by hiding the exact number of samples included:

$$\widetilde{w} = \frac{w}{\min\left(1, \frac{1}{\|w\|_2 + \epsilon}\right)} + \mathcal{N}\left(0, \sigma^2 C^2 I\right) \tag{4.3}$$

Where:

- w are the original weights,
- ϵ is a small positive quantity,
- $\|w\|_2 = \sqrt{\sum_i w_i^2}$ is the L_2 norm of the weights,
- C is the clipping threshold (maximum gradient norm),
- σ is the noise multiplier controlling the noise scale,
- $\mathcal{N}\left(0, \sigma^2 C^2 I\right)$ represents Gaussian noise with zero mean and covariance matrix $\sigma^2 C^2 I$,
- $\widetilde{w}$ are the resulting weights after clipping and noise addition.

4.5 User Interface

To ensure interaction with the local LLM and support manual inspection workflows during the testing phase, a user interface was developed using Gradio (see Figure 4.5). This interface enables users to easily input questions, view responses from the model, and inspect the full documents retrieved for question answering. The accessibility of the Gradio interface makes it convenient for both technical and non-technical users, as several metrics can be adjusted for responding, such as temperature, top-p, top-k, and number of beams. This UI played a key role in streamlining the inspection and debugging process throughout development and experimentation. As shown in Figure 4.6, after entering a question, the interface returns both the generated answer and the retrieved document. On the right side, users can view not only the content of the retrieved document but also its associated metadata, such as the author, title, and creation date. This design ensures that users can verify the generated answers directly from the source.

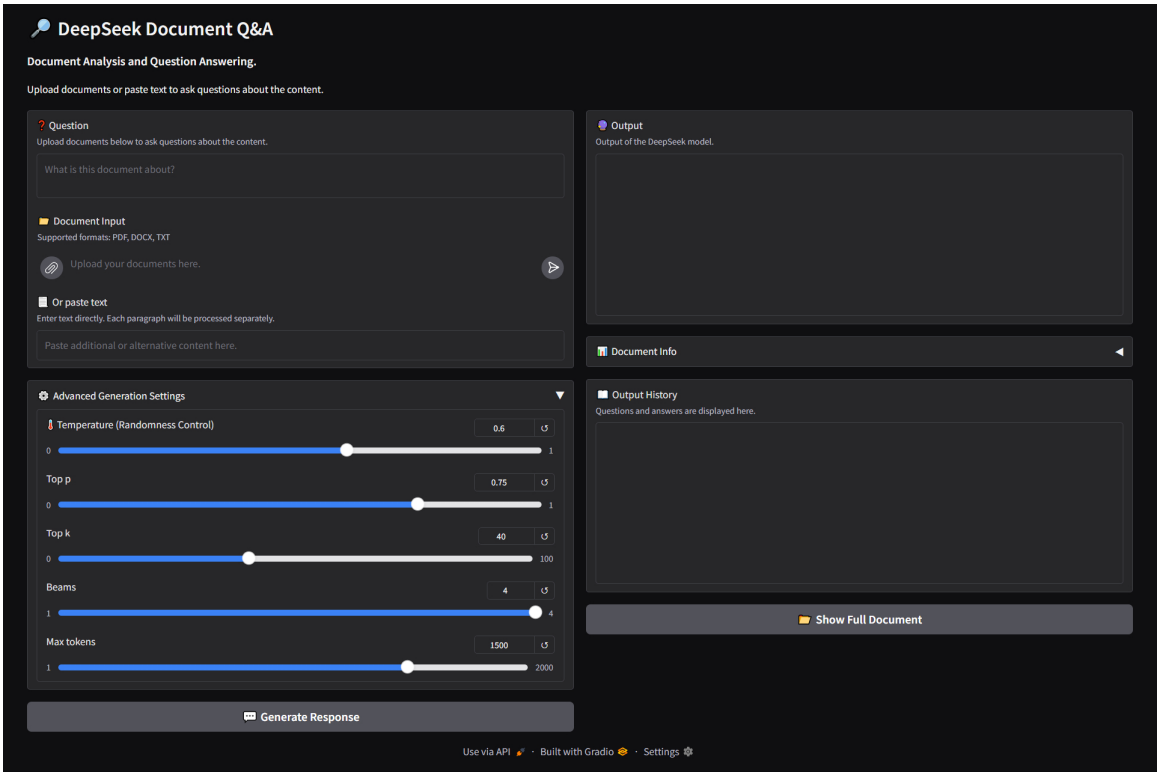

Figure 4.5: User interface built with Gradio for interacting with the local LLM. On the left the user can enter the question, document and additional plaintext context. The pannel in the bottom left corner shows the parameters for generation which can be adjusted to the users whishes. On the right the output of the LLM should be shown, which is initially empty.

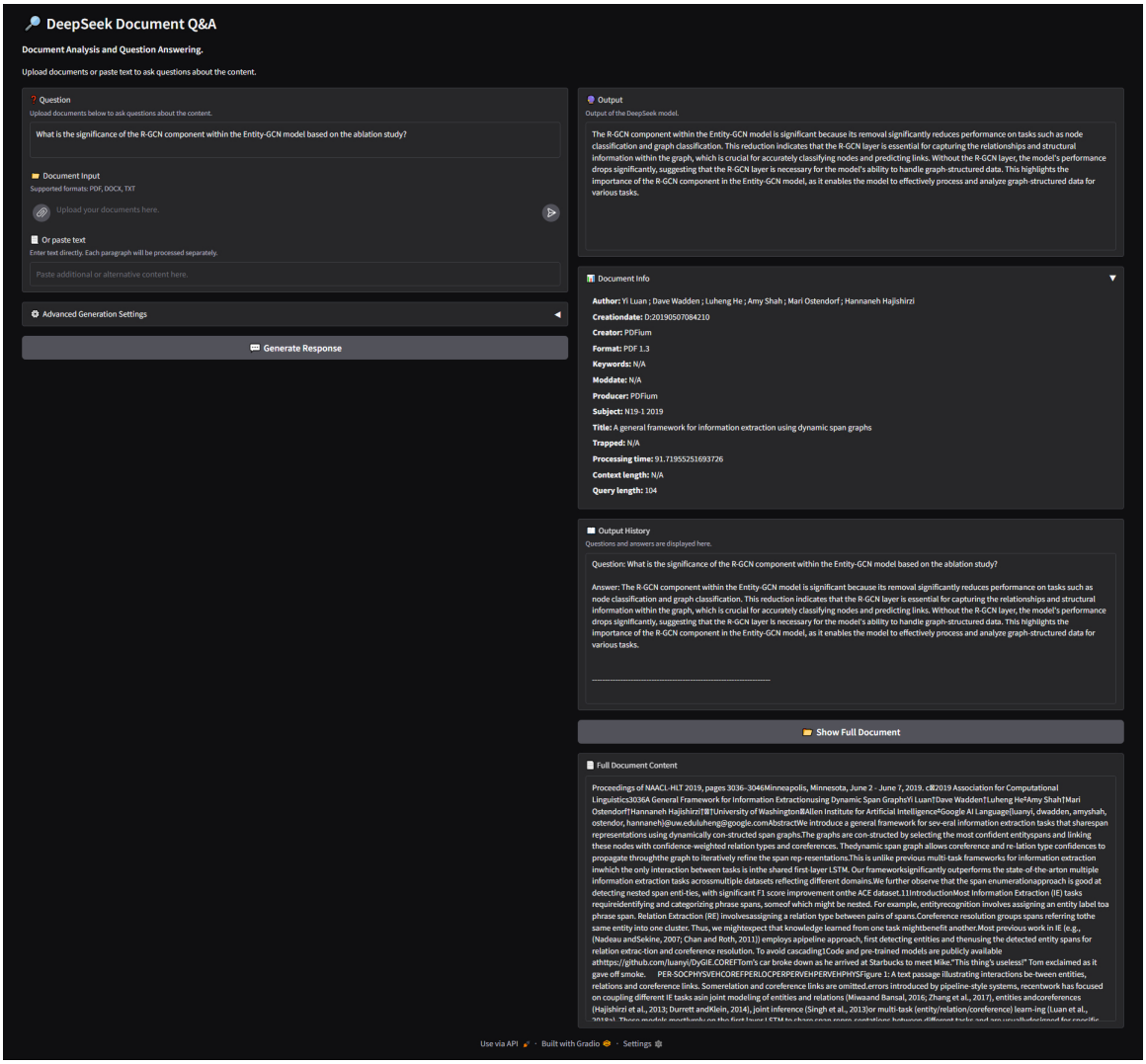

Figure 4.6: User interface displayed after submitting a query to the LLM, showing the generated answer and the retrieved document with associated metadata on the right side. From top to bottom: the first section presents the plain answer, followed by metadata details of the retrieved document, a history panel showing past queries and responses, and finally the full text of the retrieved document.

Chapter 5

Experimental Setup

This section describes the implementation of the overall secure federated LLM, including document processing and the retrieval pipeline used for question-answering over classified information. The implementation was done in Python and utilizes libraries from the LangChain framework, as well as Transformers, FAISS, TenSEAL, and Gradio for the user interface (see Figure 4.5). Concluding this section, an overview of the experiments will be provided, of which the results will be discussed in the next chapter.

5.1 Model and Dataset

5.1.1 Model Selection

In the baseline and secure framework (see section 4.3 and section 4.4), the model can be interchanged with any chosen model. `DeepSeek-R1-Distill-Qwen-1.5B` was selected as the global and local model. The temperature, top-p, top-k, and number of beams parameters can be customized based on client preferences; however, for testing purposes, the following values were used: temperature = 0.3, top-p = 0.9, top-k = 50, and number of beams = 1. Further assessment was conducted using a maximum of 2,000 new tokens to accommodate DeepSeek's reasoning process and ensure a comprehensive response.

The retrieval embedding model `BAAI/bge-small-en-v1.5` [76] was used to retrieve the top-k documents from the spaces. Using this embedding model, we observed that relevant documents were retrieved, where, based on these vectors, the embeddings returned the correct documents related to the query.

5.1.2 Dataset

The paper by Zou et al. developed with the help of human annotators and synthetic questions DOCBENCH, a benchmark for evaluating LLMs that need to process documents [77]. DOCBENCH consists of 229 real-world PDF documents and 1,102 questions across five domains: Academia, Finance, Government, Laws, and News (see Table 5.1). It features four question types, including text-only and multi-modal queries. For instance, in the Academia domain, questions may include *"Why does the model not perform as well in German compared to Spanish and Dutch?"*, in Finance, *"By how much did the number of Erica users increase from 2018 to 2019?"*, in Government, *"What is the primary focus of Bureau Objective 3.4?"*, in Laws, *"How many times does the report mention "scientific ethics"?"*, and in News, *"Is the article about Hurricane Ian's impact in Florida written by*

multiple authors?". Multi-modal questions may require combining text information with tables or figures from the PDFs, such as *"According to Figure 4, does the training speed of NCRF++ continue to increase after a batch size of 100?"*. The distribution across the different domains and question types is plotted in Figure 5.1.

Category	#Questions	Q-Tokens (K)	#Docs	#Pages	Size (KB)	D-Tokens (K)
Aca.	303	16.8	49	11	847	11.1
Fin.	288	16.8	40	192	6,594	149.4
Gov.	148	14.1	44	69	2,183	36.1
Laws	191	15.4	46	58	969	32.3
News	172	13.5	50	1	3,095	2.9
Total/Avg.	1,102	15.7	229	66	2,738	46.4

Table 5.1: Statistics per category: number of questions, question tokens (in thousands), number of documents, pages, document size in KB, and document tokens (in thousands).

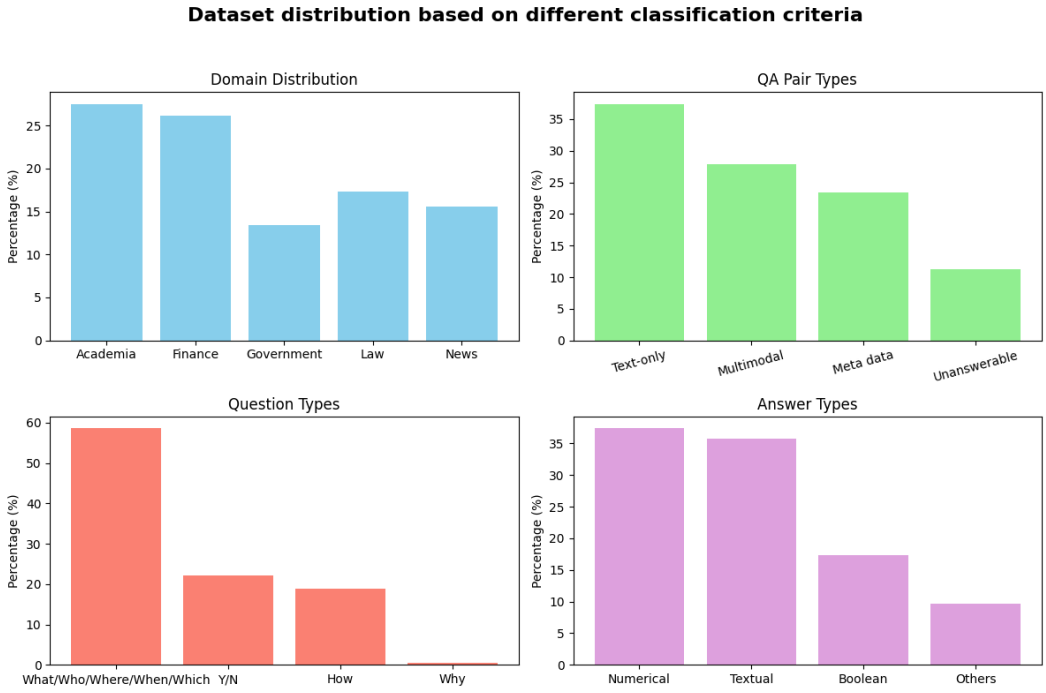

Figure 5.1: Distribution of the DOCBENCH data.

Only one dataset and a single pre-trained LLM were used across all experiments, which is sufficient within the scope of this study. The primary research objective is to explore the feasibility and security of applying FL for fine-tuning LLMs on sensitive data, rather than benchmarking model performance across different tasks or domains. This ensures that the observed results can directly be attributed to the methods being evaluated rather than variability introduced by multiple models or heterogeneous data sources. The DOCBENCH dataset structure is appropriate for evaluating the effectiveness of retrieval, policy enforcement, and local model reasoning under permission constraints. DeepSeek

is representative of several open-source LLMs that are large enough to demonstrate reasoning and answering capabilities but small enough to be deployed in resource-constrained offline environments. Introducing multiple datasets or models would significantly increase computational complexity and experimental variance, which would limit the effects of the security on the proposed framework.

5.1.3 System Specifications

The experiments are conducted using the following hardware configuration: an Intel Xeon 4214 CPU, an NVIDIA RTX 2080 Ti GPU with 11 GB of VRAM, and 16 GB of system RAM. The operating system is Ubuntu 22.04.1 running with the newest kernel. Additionally, Google Colab was used for faster and more flexible testing using Ollama (also see section 5.4.1).

5.2 Data Pre-processing

As the dataset consists of multiple PDF files with questions in JSON format, some pre-processing is applied to generate the training and test data. A custom dataset is created from these PDF files, where a random space index is assigned to each document and a corresponding question for instruction tuning. Using the PyMuPDF library, the content and metadata can be extracted from the files and added to the custom dataset. A data sample in the dataset consists of a question, the document context, the ground truth, the space key index, and metadata.

5.2.1 Test Set Selection

A distinct set of questions was reserved for testing, which ensures that there was no overlap between the questions used for fine-tuning each client's local LLM. For each of the space key indices, five questions were chosen (see also Table 9.1). A summary is given of the selected questions after random sampling. A manual inspection is done after sampling, as questions such as *"Who is the last author of the paper?"* and *"How many words are there in total in the paper?"* are impractical because the model cannot link such questions to an exact document. It is not clear which paper is meant in these questions, as the model will answer with the relevant paper or document; otherwise, the model cannot generate a reliable answer. These questions, which query the model for knowledge about the metadata of the documents, were replaced with more broadly answerable questions. Ultimately, the final test set contains five questions for each space key, ensuring a balanced combination of randomness and practical answerability.

5.3 Federated Learning Setup

The following section outlines how permission enforcement is implemented in the federated setup, including the structure and roles defined by global and space-specific permissions.

5.3.1 Permission Enforcement

The retrieval framework follows an RBAC model to enforce permissions during document retrieval. Furthermore, permissions are divided into two categories: global and space permissions [44]. The first are site-wide permissions, which can be granted to a user or a group, and are assigned by an administrator. These permissions encompass the types that govern whether a user can log in or create a space. However, they do not interact with space permissions or page restrictions.

Every space has its own independent set of permissions, managed by the space-level RBAC permission rules. This set of permissions can grant or revoke the permissions described in Table 5.2. Revoking permissions can be done on a per-user or per-group basis, ensuring that document retrieval respects the assigned roles. From these categories, it becomes clear that permissions are not granted directly to the individual clients but rather assigned based on their roles or groups, which is the core principle of RBAC.

Permission Name
Viewspace Permission
Remove Own Content Permission
Create/Edit Page Permission
Remove Page Permission
Edit Blog Permission
Remove Blog Permission
Create Attachment Permission
Remove Attachment Permission
Comment Permission
Remove Comment Permission
Set Page Permissions Permission
Remove Mail Permission
Export Space Permission
Administer Space Permission

Table 5.2: Space permissions based on Confluence.

Page restrictions operate in addition to the RBAC hierarchy. Pages are open for viewing and editing by default; however, it is possible to restrict either viewing or editing to specific users or groups. Every page is part of a space, and space permissions allow the space admin to revoke permissions to view or edit content for the whole space.

Confluence Implementation

The goal is to replicate Confluence's space-level permission model, where users can view, edit, comment, export, and have admin rights, and which supports viewing and cloning these permissions across users, based on the Java approach in [78]. This repository provides a REST-based interface for managing space-level user permissions for Confluence. The architecture enables cloning or replicating permissions to another user; the system offers a PUT endpoint that accepts the permissions data structure and applies it to the selected target user. Using different managers, such as `RestUserPermissionManager`, `UserPermissionsResource`, `UserManager`, and `SpaceManager`, the permissions for the spaces can be managed to ultimately obtain the permissions for the users. Leveraging Confluence's widespread adoption ensures that this model aligns with real-world collaborative environments and provides a practical framework for managing permissions within an enterprise.

5.3.2 Clients

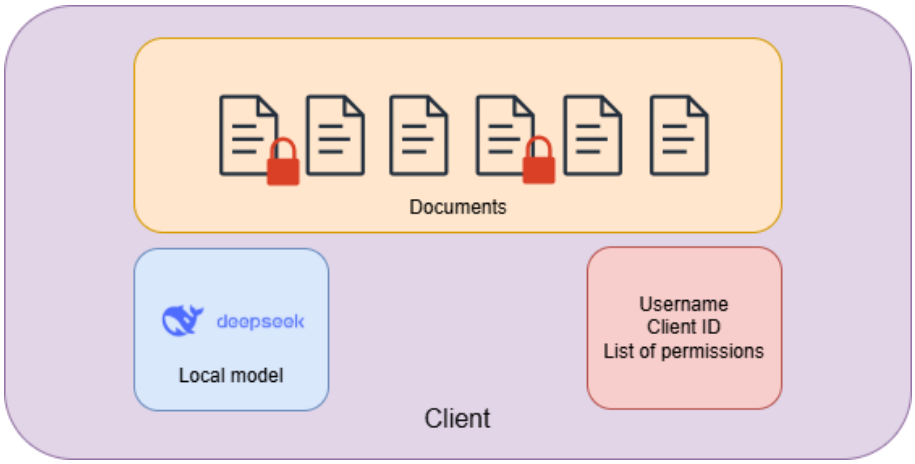

Figure 5.2: Visual representation of the client where each client has access to one or more spaces, which have documents in them. The client has their local LLM, username, client ID, and list of permissions per space.

Each client maintains:

- A local copy of the LLM (DeepSeek-R1), downloaded from a central global model server using PEFT.
- A personalized subset of documents filtered according to the client's specific permissions.
- A user profile consisting of a username, accessible documents, and a list of permissions, such as a Confluence space.

The client can ask questions to the local model, which is used for local model updates only, and the global model is utilized for sending these updates after aggregation. This ensures privacy, but the client only gets answers based on its subset of the documents. This document store on each device is indexed in a local vector space using FAISS [79] to enable semantic retrieval when querying. When the client issues a question, the client embeds the query locally and retrieves the top-k relevant documents from its index. These results are concatenated to the prompt template and passed to the LLM, which generates an answer grounded only on the user's authorized documents. Before uploading the parameters to the global model, each client encrypts its updates using TenSEAL's CKKS scheme. The CKKS scheme was configured with a polynomial modulus degree of 32768 and coefficient modulus sizes of $[60, 40, 40, 60]$, which determines the precision and security of computations on encrypted data using a chain of modulus primes [80]. This setup ensured the secure aggregation of model updates, sharing only the public key with the server while maintaining the confidentiality of the client's private data. The weights of the local model are then shared with the global model, and updates are sent to all local models, allowing each client to benefit from the globally improved model while maintaining its data privacy. Clients can decrypt the updates after aggregation and integrate them into their local models to use for inference.

Each client has a local state, which consists of its username, the set of indexed documents filtered by space-level permissions, and a local DeepSeek instance that is fine-tuned on their instructions. Table 5.3 summarizes the mock users, their spaces, and their assigned roles.

ID	Name	Spaces	Is Admin	Permissions/Role
1	admin	[0, 1]	Yes	admin
2	user1	[1, 2]	No	editor
3	user2	[2]	No	viewer
4	user3	[3]	Yes	editor
5	user4	[0]	No	viewer
6	user5	[1, 3]	No	editor
7	user6	[2]	No	editor
8	user7	[0, 3]	Yes	viewer
9	user8	[1]	No	viewer
10	user9	[3]	No	editor
11	user10	[0, 1, 2, 3]	Yes	admin

Table 5.3: Client Configuration in the Federated Learning Setup, Including Role Assignments, Accessible Spaces, and Admin Status

5.3.3 Global Server

The server starts by distributing the global model to the selected clients. Each client loads the global model into their local model, from which the fine-tuning process can begin. After training on the private local dataset of the clients, encrypted parameters are sent back to the server, where the weights are aggregated. The server only has access to the number of clients; it has no additional information.

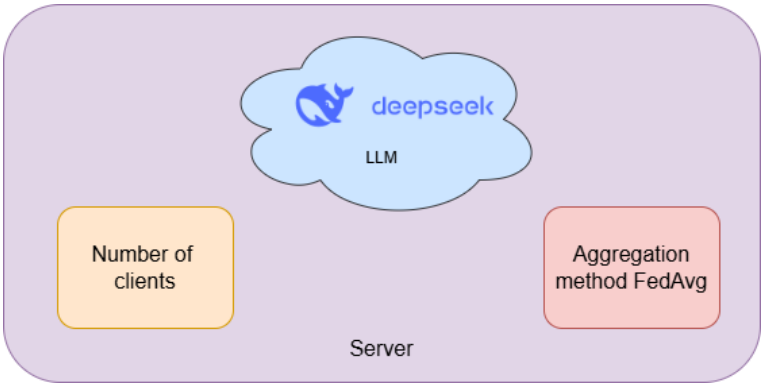

Figure 5.3: Global server

5.3.4 Training Process

In the experimental setup, training occurs in a series of communication rounds, where each round mimics the real-world communication between the server and clients. At the start of each round, a random subset of clients is chosen to participate in the training process. After loading the global model, the fine-tuning or training process can begin with unseen documents, incorporating new instructions from the client. Before training, the data were divided into training and test sets, with a ratio of 70%/30%. To reduce computational and communication burdens, PEFT techniques are employed, specifically LoRA with rank $r = 16$ and alpha $\alpha = 16$. This configuration optimized communication and computation. The fine-tuning is performed using Adam optimizer with a learning rate of $1e - 2$ for 1 epoch. Clients train with a batch size of 2 and retrieve relevant documents

using the FAISS index. Algorithm 1 represents the secure fine-tuning process, including the training process.

Algorithm 1 Federated Secure LLM Training Procedure

Input: N: Total number of clients, R: Communication rounds, D: Client datasets, m_0: Initial global model, C: CKKS encryption context
Output: Final model weights m_F, training loss history $\mathcal{L}$
Initialize global model weights m_0
Initialize encryption context C (CKKS scheme)
for each round $r = 1$ to R **do**
    Randomly select a subset of clients $\mathcal{S}_r \subseteq N$
    Initialize server with current global model m_r
    for each client $i \in \mathcal{S}_r$ **do**
        Download and set global model w_r locally
        Prepare client-specific dataset D_i based on access permissions
        Tokenize and preprocess local data
        Initialize training configuration with chosen hyperparameters
        Train the local model for a fixed number of epochs
        Record training loss ℓ_i^r for analysis
        Encrypt the updated local model weights using context C
        Save encrypted updates for server aggregation
    end for
    Perform secure aggregation of encrypted weights across selected clients
    Decrypt the aggregated model using the private key
    Update global model weights: $w_{r+1} \leftarrow w_r + \Delta w_r$
    Store updated model and training loss
end for
return Final model weights m_F, training loss history $\mathcal{L}$

5.4 Experiments

The experiments should assess the accuracy and security of the proposed FL framework for permission handling.

5.4.1 Evaluation Metrics

IXN Scores

To evaluate retrieval security, we focus on whether the framework enforces access control during retrieval by conducting experiments in a controlled environment with setup clients and their corresponding permissions to several spaces. These retrieved documents will be compared against the baseline (see section 4.3), which retrieves the top-k documents without enforcing any retrieval permissions. To measure information loss due to access restrictions, the IXN score is used for each user u_i:

$$IXN_{u_i} = \frac{|D^{fi} \cap D^{fs|u_i}|}{|D^{fi}|} \tag{5.1}$$

where D are the documents, fi baseline federated learning and fs secure federated learning [53]. A low IXN score indicates more restrictive access, while higher scores indicate more permissive access privileges. Users with more permitted spaces are expected to achieve a higher IXN score.

Response Quality

ROUGE-1, ROUGE-L, and BLEU are employed to evaluate the generated response quality on question and answer tasks using NLP. Since the LLM generation step is identical between both the baseline and secure variants of FL, significant deviations in these scores indicate that access-controlled retrieval is altering the quality of the response. The ROUGE metric can be considered a measure of how much of the baseline response content is also present in the target response. ROUGE-1 measures the number of matching n-grams between the model-generated text and ground truth answer, while ROUGE-L captures the length of the longest common subsequence shared between texts [81]. ROUGE-1 computes the portion of n-grams, in this case unigrams, present in the ground truth with:

$$\text{ROUGE-1} = \frac{\sum_{gram_1 \in S} \text{Count}_{match}(gram_1)}{\sum_{gram_1 \in S} \text{Count}(gram_1)} \tag{5.2}$$

where S is the set of retrieved documents.

On the other hand, BLEU estimates the precision by counting the n-grams in the generated answer that overlap with the ground truth [82]. The metric has a range from 0 to 1, where 1 means an almost identical response compared to the ground truth. However, almost similar answers will receive a score lower than one unless they are identical to the ground truth. This also results in the human answer not necessarily scoring 1. The overall BLEU score consists of a brevity penalty BP, which penalizes the answers with perfect precision but without meaning:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{5.3}$$

Then the overall BLEU score will be computed as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{5.4}$$

These scores are NLP metrics that match words between the generated LLM answer and the ground truth, rather than evaluating the answers based on their semantic content. Therefore, the following metric offers a method for evaluating semantics as well.

RAGAS

RAGAS is suited for reference-free evaluation, as it offers several distinct, LLM-driven metrics that assess different characteristics of RAG without the need for golden documents or ground-truth labels. It utilizes an LLM judge, where calls to the judge are made to evaluate the model's output.

While DeepSeek is the base model for this FL setup, it was not used as the evaluation judge since RAGAS requires a LLM capable of providing consistent reference-free judgements. `llama3:8b` offers a strong balance of generation quality and efficiency, which is used as the LLM for evaluation running locally using the Ollama server. As we are interested in assessing the federated LLMs along the components of retrieval relevancy to the query and generation response quality, the following RAGAS metrics were selected [83]:

1. **Context Precision:** Context precision measures the proportion of relevant chunks in the retrieved documents. It will reveal whether the LLM is returning unnecessary noise or redundant data at the retrieval stage. It is computed as the mean of the precision for each chunk in the context, which is the ratio of the number of relevant chunks to the total number of chunks:

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \cdot v_k)}{\text{Total number of relevant items in the top K results}} \tag{5.5}$$

where

$$\text{Precision@k} = \frac{\text{true positives@k}}{(\text{true positives@k+false positives@k})} \tag{5.6}$$

and K is the total number of chunks and $v_k \in \{0, 1\}$ is the indicator for relevance at rank k [84]. If the precision is low, the model is retrieving irrelevant documents. It is crucial to minimize the retrieval of irrelevant data to prevent privacy leakage or security violations.

2. **Answer Relevancy:** Answer relevancy quantifies how closely the generated answer aligns semantically with the original query. When building these systems for clients, the answers must be relevant to the client's query. Answers that are incomplete or contain unnecessary information will receive a lower score, whereas answers with better relevance will get higher scores. The judge LLM generates multiple synthetic questions from the answer, then each generated question embedding is compared using cosine similarity to the original query embedding:

$$\text{Answer Relevancy} = \frac{1}{N} \sum_{i=1}^{N} \cos(E_{g_i}, E_q) \tag{5.7}$$

where E_{g_i} is the embedding of the generated question i. E_q is the embedding of the original question. N is the number of generated questions [85].

3. **Faithfulness:** Faithfulness assesses whether the claims made in the generated response can be found in the retrieved documents. Evaluating whether the made claim is in the answer is supported by the retrieved document, and will detect if there is any hallucination present in the LLM-generated response. When generating answers that contain sensitive information, there must be a guarantee that the outputs are based on these sensitive documents. The judge LLM extracts statements from the model's answer and verifies each one against. It has a range of 0 to 1, where higher scores indicate better consistency. The score is the fraction of verified claims [86]:

$$\text{Faithfulness score} = \frac{\text{Number of claims in the response supported by the retrieved context}}{\text{Total number of claims in the response}} \tag{5.8}$$

4. **Context Recall:** The last metric, context recall, measures how many of the documents were successfully retrieved [87]. High context recall means that the retrieved documents contain most or all of the useful information. Using the client's input, the reference, and the retrieved context, the value computed ranges between 0 and 1. It measures whether the model retrieved enough relevant information to fully answer the question. This metric is not enough to evaluate retrieval on its own, so it needs to be combined with precision and faithfulness. The formula for context recall uses the number of claims or facts:

$$\text{Context Recall} = \frac{\text{Number of claims in the reference supported by the retrieved context}}{\text{Total number of claims in the reference}} \tag{5.9}$$

where the reference is the ground truth answer from the dataset.

Chapter 6

Results

The federated secure LLM will be evaluated based on the quality of generated responses and the security guarantees provided during the retrieval step. The following two frameworks will be considered (see section 4.3 and section 4.4):

- **Insecure federated LLM** (*baseline*): standard query answering without permission enforcement. There is retrieval across all decentralized spaces containing classified documents.
- **Secure federated LLM**: federated query answering with permission enforcement and extra security measures enforced during the FL process. Retrieval results in an authentication process, ensuring the secure retrieval of the top documents among permitted documents.

The primary difference between the two frameworks is that the secure LLM utilizes authorization filters to enforce access control. To adequately measure the performance of both architectures, the security access control mechanism is tested to detect any leaking information and compare this with the baseline.

6.1 Training Results

During the training of each communication round, the average loss for 10 epochs of each client was saved and is plotted in Figure 6.1. The loss starts high with an average of 1.35 and reaches a loss of 0.37 after 10 rounds of FL. The results did not converged after these 10 rounds. In Table 6.1, the number of rounds in which the clients participated is displayed. For each round of communication they participated in, the training loss is visualized in Figure 6.2. Not every client participated in the same number of rounds, thereby reflecting the irregular participation patterns that are typical in a real-world FL environment.

Name	Number of Rounds Participated
admin	3
user 1	3
user 2	6
user 3	2
user 4	6
user 5	4
user 6	1
user 7	3
user 8	4
user 9	5
user 10	3

Table 6.1: Number of communication rounds each client participated in during the FL process.

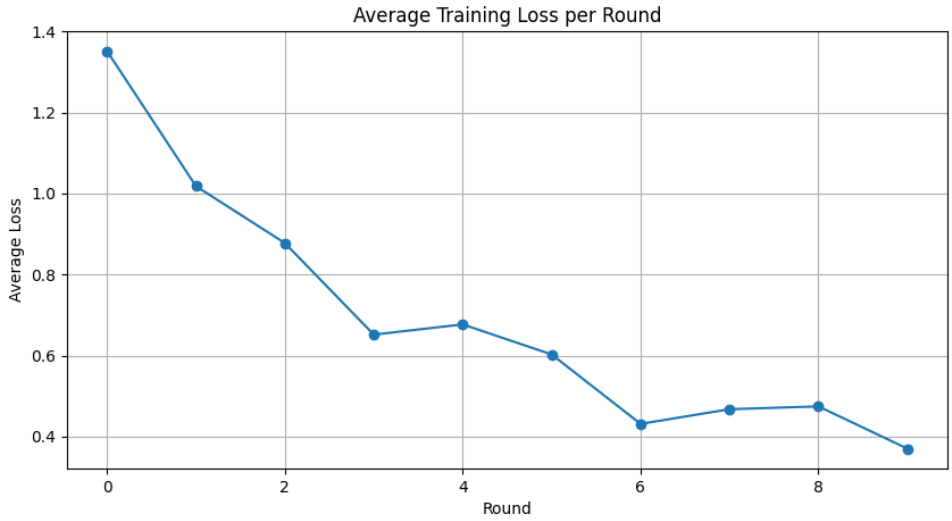

Figure 6.1: Average training loss over 10 rounds in the proposed federated learning setup. The plot shows the average training loss across all participating clients per round.

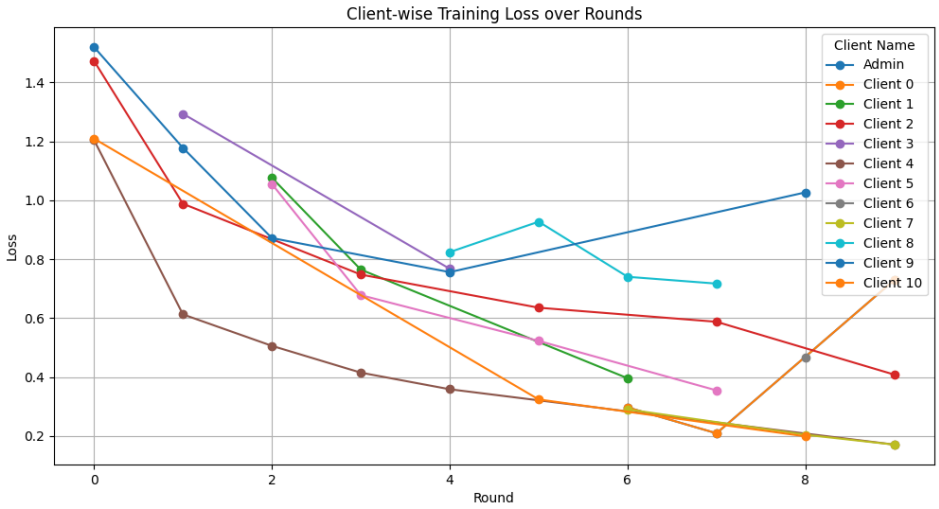

Figure 6.2: Training loss progression for each individual client, highlighting variations in learning dynamics. Only the rounds where the client participated in has a data point, the line is connecting the participating rounds for each client.

6.2 IXN score

To evaluate the effectiveness of the proposed RBAC-based retrieval system, which is based on permissions from Confluence, the IXN scores are computed for the chosen clients, each assigned access to spaces defined as in Table 5.3. The IXN score is defined as the intersection between the top 10 documents of the secure and insecure RAG embeddings. These results are presented in Table 6.2. For each user, the average score is computed, where lower scores should indicate that access control policies successfully restricted user access to one or more top documents retrieved by the baseline framework [53]. The results vary significantly across user roles, with users 2, 3, 6, and 9 achieving the highest average IXN score of 0.22 compared to the baseline. This high IXN score indicates that the retrieved documents from these users are for 22% similar to the documents retrieved by the insecure baseline. In contrast, users 4, 7, and 10, as well as the admin, demonstrate more restricted document access, with an average of 0.12. These lower scores indicate less overlap with the retrieved baseline documents. Users 1, 5, and 8 retrieved a score of 0.19 which is lower than the highest scoring users. Overall are the scores low, which was not hypothesized as some of the documents should have been retrieved according to the permission levels of the clients. The reason for these lower scores will be discussed in chapter 7. We cannot validate, according to this definition of the IXN scores, that permission enforcement was successful.

Question	Baseline	admin	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
1	1	0	0.1	0.2	0.1	0	0.1	0.2	0	0.1	0.1	0
2	1	0.2	0.1	0.1	0.2	0.2	0.1	0.1	0.2	0.1	0.2	0.2
3	1	0.1	0.3	0.2	0.3	0.1	0.3	0.2	0.1	0.3	0.3	0.1
4	1	0.5	0.2	0.1	0.2	0.5	0.2	0.1	0.5	0.2	0.2	0.5
5	1	0.2	0	0.2	0.2	0.2	0	0.2	0.2	0	0.2	0.2
6	1	0.1	0.3	0.2	0.4	0.1	0.3	0.2	0.1	0.3	0.4	0.1
7	1	0.2	0.4	0.3	0.1	0.2	0.4	0.3	0.2	0.4	0.1	0.2
8	1	0	0.3	0.3	0.3	0	0.3	0.3	0	0.3	0.3	0
9	1	0.2	0.1	0	0.2	0.2	0.1	0	0.2	0.1	0.2	0.2
10	1	0.1	0.1	0.2	0.2	0.1	0.1	0.2	0.1	0.1	0.2	0.1
11	1	0	0.2	0.3	0.4	0	0.2	0.3	0	0.2	0.4	0
12	1	0.1	0.3	0.3	0	0.1	0.3	0.3	0.1	0.3	0	0.1
13	1	0.1	0.1	0.1	0	0.1	0.1	0.1	0.1	0.1	0	0.1
14	1	0	0.2	0.2	0.6	0	0.2	0.2	0	0.2	0.6	0
15	1	0.1	0.1	0.3	0.5	0.1	0.1	0.3	0.1	0.1	0.5	0.1
16	1	0	0.1	0.6	0.1	0	0.1	0.6	0	0.1	0.1	0
17	1	0.2	0.3	0.1	0.3	0.2	0.3	0.1	0.2	0.3	0.3	0.2
18	1	0.1	0.3	0.2	0.1	0.1	0.3	0.2	0.1	0.3	0.1	0.1
19	1	0	0.3	0.1	0.2	0	0.3	0.1	0	0.3	0.2	0
20	1	0.2	0.1	0.4	0	0.2	0.1	0.4	0.2	0.1	0	0.2
Average		0.12	0.19	0.22	0.22	0.12	0.19	0.22	0.12	0.19	0.22	0.12

Table 6.2: IXN Results where the set intersections are computed between the retrieved documents of the baseline and the users. All users are compared against the baseline, which did not contain any permission enforcements on the documents when querying the questions.

6.3 Response Quality

The average BLEU, ROUGE-1, and ROUGE-L scores across the 20 questions are summarized in Table 6.3 with the highest and lowest scores for each metric column highlighted in yellow and pink, respectively. Two users, user 2 (ROUGE-1-P = 0.0816, ROUGE-L-P = 0.0771) and 3 (ROUGE-1-R = 0.4387, ROUGE-L-R = 0.3912), received the highest scores for two of the metrics. User 1 outperforms the baseline in BLEU with a score of 0.0893, suggesting that this user did not negatively impact fluency when answering. In ROUGE-1 precision, user 2 stands out with the highest score of 0.0816, indicating that the outputs include more relevant n-grams relative to their shorter answers. This suggests a higher precision at the cost of recall. Meanwhile, user 3 achieves the highest ROUGE-1 recall, implying that the answers cover a greater portion of the given context. ROUGE-L scores show a similar pattern, as the baseline also has the best overall ROUGE-L-F score of 0.1052, but certain users excel in other metrics. For instance, user 2 has the highest ROUGE-L precision score (0.0771), and user 3 again has the highest recall (0.3912). User 10 achieves the best ROUGE-L-F score among all users, indicating a good balance between incorporating context and generating a coherent output.

User 1, despite having the highest BLEU score, exhibits the lowest ROUGE-1 precision, F1 score, and ROUGE-L precision and F1 scores. This suggests that while the output may closely match specific exact phrases, it covers only a small portion of the full referenced context, indicating low

completeness. Similarly, user 2 has the lowest ROUGE-1 and ROUGE-L recall, indicating that the generated answers are either very short or miss substantial portions of relevant information from the given context.

The same results are plotted in Figure 6.3, where the baseline stands out and receives a higher score on almost every response quality metric.

Name	BLEU	ROUGE-1-P	ROUGE-1-R	ROUGE-1-F	ROUGE-L-P	ROUGE-L-R	ROUGE-L-F
baseline	0.0863	0.0849	0.4099	0.1256	0.0700	0.3618	0.1052
admin	0.0417	0.0476	0.4138	0.0783	0.0423	0.3745	0.0701
user 1	**0.0893**	**0.0205**	0.3919	**0.0376**	**0.0176**	0.3409	**0.0322**
user 2	0.0598	**0.0816**	**0.3054**	0.0554	**0.0771**	**0.2691**	0.0482
user 3	0.0850	0.0254	**0.4387**	0.0444	0.0223	**0.3912**	0.0388
user 4	0.0721	0.0260	0.4191	0.0454	0.0219	0.3738	0.0384
user 5	0.0628	0.0227	0.4147	0.0389	0.0206	0.3749	0.0351
user 6	0.0479	0.0316	0.4023	**0.0486**	0.0263	0.3525	0.0405
user 7	0.0513	0.0232	0.3525	0.0402	0.0213	0.3122	0.0369
user 8	0.0501	0.0272	0.3971	0.0454	0.0220	0.3513	0.0364
user 9	**0.0398**	0.0261	0.4240	0.0404	0.0239	0.3896	0.0368
user 10	0.0556	0.0415	0.3637	0.0548	0.0327	0.3387	**0.0458**

Table 6.3: The average BLEU, ROUGE-1 (precision, recall, F1), and ROUGE-L (precision, recall, F1) scores for each user compared to a baseline. It highlights the performance of user-generated responses in terms of fluency and relevance. The highest and lowest scores for each response quality metric are highlighted in yellow and pink, respectively.

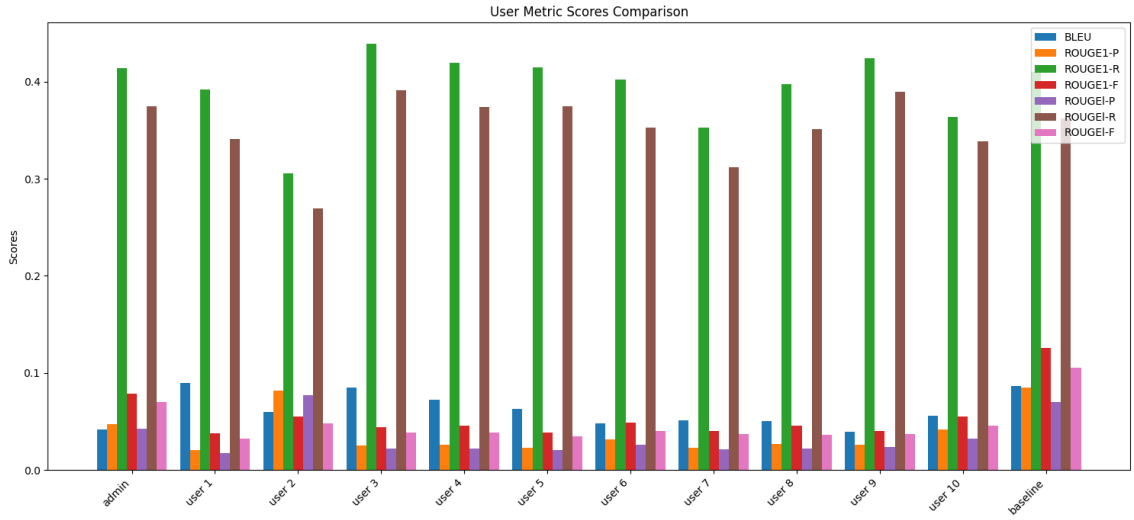

Figure 6.3: Comparison of BLEU and ROUGE metric scores across different users, illustrating the variations in text generation quality and overlap with reference texts.

To better assess how document access policies influence the quality of RAG of each user for each of the spaces, Table 6.4 summarizes the mean of various BLEU and ROUGE scores across the individual space keys for each user, with the highest score for each metric highlighted in pink and the accessible spaces for each user highlighted in blue. For most users, the space key they have access to stands out, consistently achieving higher scores across various metrics. For admin, user 1, 3, 5, 6, 7,

and 10, the results are the highest within the space key(s) they have access to. The other users have higher scores for other space keys that they do not have access to.

Name	Space Key	BLEU	ROUGE-1-P	ROUGE-1-R	ROUGE-1-F	ROUGE-L-P	ROUGE-L-R	ROUGE-L-F
admin	**0**	0.0346	**0.0618**	0.2696	0.0905	0.0482	0.2333	0.0708
	1	**0.0521**	0.0574	**0.6234**	**0.1050**	**0.0564**	**0.5574**	**0.0964**
	2	0.0464	0.0614	0.5612	0.0994	0.0554	0.5121	0.0959
	3	0.0338	0.0100	0.2012	0.0184	0.0093	0.1951	0.0171
user 1	0	0.0495	0.0131	0.3542	0.0247	0.0098	0.2817	0.0184
	1	**0.2156**	**0.0361**	**0.5611**	**0.0657**	**0.0311**	**0.5011**	**0.0567**
	2	0.0416	0.0174	0.4028	0.0313	0.0170	0.3821	0.0305
	3	0.0506	0.0155	0.2497	0.0288	0.0126	0.1988	0.0233
user 2	0	0.0449	0.0501	0.3063	0.0759	0.0396	0.2536	0.0595
	1	**0.0956**	**0.2089**	0.3327	0.0406	**0.2079**	**0.3145**	0.0388
	2	0.0259	0.0154	**0.3341**	**0.0780**	0.0143	0.2975	**0.0695**
	3	0.0729	0.0522	0.2485	0.0273	0.0466	0.2109	0.0252
user 3	0	0.0266	0.0118	0.3546	0.0227	0.0101	0.3180	0.0193
	1	0.0374	0.0084	0.4497	0.0162	0.0065	0.4061	0.0127
	2	0.0512	0.0353	0.4086	0.0643	0.0272	0.3188	0.0496
	3	**0.2247**	**0.0460**	**0.5420**	**0.0746**	**0.0455**	**0.5218**	**0.0735**
user 4	**0**	0.0306	**0.0428**	0.3264	**0.0688**	**0.0328**	0.2821	**0.0531**
	1	**0.1556**	0.0198	**0.5829**	0.0377	0.0165	**0.5354**	0.0315
	2	0.0216	0.0109	0.3784	0.0206	0.0090	0.3134	0.0169
	3	0.0807	0.0304	0.3885	0.0545	0.0292	0.3642	0.0521
user 5	0	0.0408	0.0138	0.3301	0.0260	0.0109	0.2859	0.0205
	1	**0.1052**	**0.0621**	**0.6076**	**0.1004**	**0.0590**	**0.5735**	**0.0952**
	2	0.0475	0.0096	0.4470	0.0187	0.0079	0.3965	0.0154
	3	0.0577	0.0053	0.2739	0.0104	0.0046	0.2436	0.0090
user 6	0	0.0209	0.0336	0.4905	0.0485	0.0272	0.4459	0.0391
	1	0.0687	0.0274	0.3730	0.0417	0.0203	0.3265	0.0311
	2	**0.0689**	**0.0539**	**0.5203**	**0.0831**	**0.0489**	**0.4569**	**0.0753**
	3	0.0332	0.0114	0.2254	0.0211	0.0089	0.1806	0.0165
user 7	**0**	**0.0737**	**0.0607**	**0.5254**	**0.1001**	**0.0562**	**0.4638**	**0.0930**
	1	0.0637	0.0081	0.2707	0.0157	0.0069	0.2304	0.0134
	2	0.0214	0.0188	0.3595	0.0349	0.0176	0.3243	0.0326
	3	0.0463	0.0052	0.2546	0.0101	0.0045	0.2303	0.0088
user 8	0	0.0310	0.0272	0.4901	0.0454	0.0208	0.4334	0.0347
	1	**0.0825**	0.0249	**0.5879**	0.0459	0.0211	**0.5370**	0.0390
	2	0.0227	**0.0416**	0.2837	**0.0639**	**0.0343**	0.2263	**0.0510**
	3	0.0641	0.0151	0.2267	0.0265	0.0118	0.2085	0.0209
user 9	0	0.0322	0.0115	0.5502	0.0220	0.0095	0.5019	0.0181
	1	**0.0636**	0.0191	**0.5886**	0.0370	0.0174	**0.5454**	0.0337
	2	0.0473	**0.0679**	0.3742	**0.0909**	**0.0633**	0.3582	**0.0851**
	3	0.0160	0.0061	0.1830	0.0118	0.0053	0.1527	0.0103
user 10	**0**	0.0266	0.0137	0.3425	0.0260	0.0116	0.3020	0.0221
	1	**0.1560**	**0.1205**	**0.5005**	**0.1345**	**0.0883**	**0.4573**	**0.1044**
	2	0.0244	0.0160	0.3504	0.0297	0.0150	0.3344	0.0278
	3	0.0155	0.0158	0.2612	0.0290	0.0158	0.2612	0.0290

Table 6.4: The mean BLEU and ROUGE scores across individual space keys for each user. The accessible spaces for each user are highlighted in blue, and the best-performing space per user (based on each metric) is additionally emphasized in pink. This provides insight into how access to different documents from the spaces affects RAG quality.

6.4 RAGAS

Table 6.5 compares the mean response quality scores for the baseline and users across the four metrics: Context Precision, Answer Relevance, Faithfulness, and Context Recall. The baseline achieves the highest scores for Context Precision (0.5789) and Context Recall (0.5125), indicating that it uses context more accurately than other users. The baseline also performs well in Answer Relevance (0.6457) but scores relatively low on Faithfulness (0.1167). Among the users, user 1 has the highest Context Precision of 0.4000 and Context Recall of 0.4750, indicating accurate and complete context retrieval. User 6 surpasses all other users, including the baseline with the highest Answer Relevance of 0.8435. Nevertheless, the Context Precision is very low (0.0526) for user 6, while user 7 has the lowest Faithfulness (0.1026), and user 4 has the lowest Context Recall (0.3267). This indicates that some users retrieve less accurate or incomplete context or provide less faithful answers.

The same results are plotted in Figure 6.4, where the baseline again stands out and receives a higher score on almost every RAGAS metric.

Name	**Context Precision**	**Answer Relevance**	**Faithfulness**	**Context Recall**
baseline	0.5789	0.6457	0.1167	0.5125
admin	0.2500	0.5882	0.1682	0.3469
user 1	**0.4000**	0.5868	0.2019	**0.4750**
user 2	**0.0526**	0.7268	0.1523	0.3868
user 3	0.1579	0.6431	**0.2706**	0.4083
user 4	0.2000	0.6468	0.1559	**0.3267**
user 5	0.3500	0.8090	0.1942	0.4292
user 6	**0.0526**	**0.8435**	0.1290	0.3335
user 7	0.2000	0.7197	**0.1026**	0.3443
user 8	0.3500	**0.5624**	0.2467	0.3450
user 9	0.1579	0.6267	0.2161	0.3958
user 10	0.2500	0.7061	0.1568	0.3589

Table 6.5: The mean response quality scores for each user and baseline across four metrics: Context Precision, Answer Relevance, Faithfulness, and Context Recall. Highlighted yellow values indicate the highest score achieved per metric, pink values are the lowest scores.

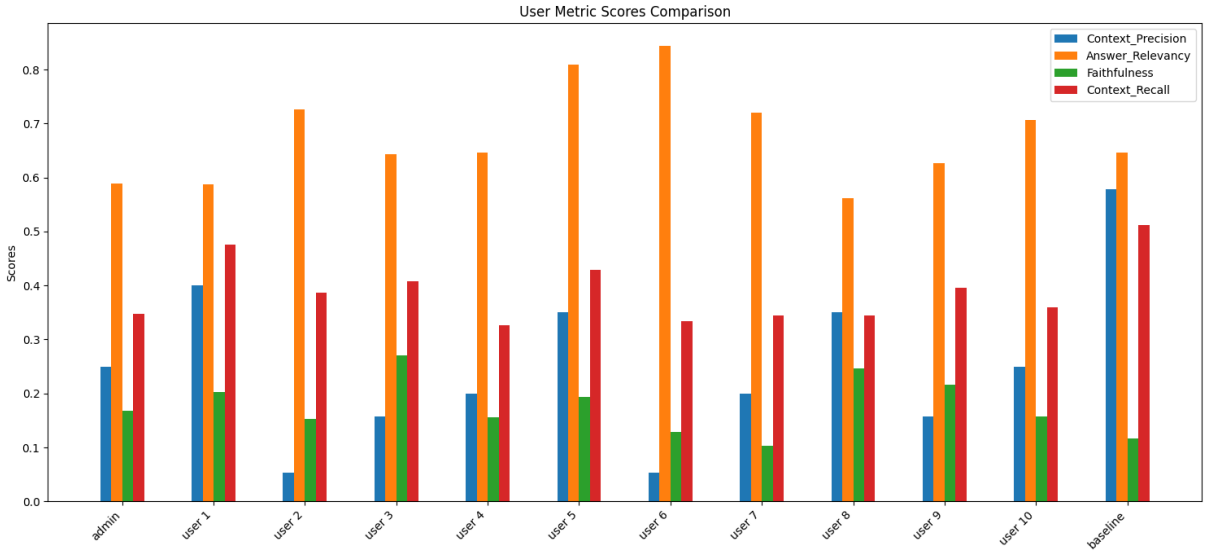

Figure 6.4: Evaluation of response quality based on the RAGAS metrics: Context Precision, Answer Relevancy, Faithfulness, and Context Recall across different users, highlighting variations in information accuracy and relevance.

Table 6.6 illustrates the relationship between user access to their spaces and performance across four metrics: Context Precision, Answer Relevance, Faithfulness, and Context Recall. Each user performs best in some of the spaces they have access to, as highlighted in blue, and the highest scores for each metric are highlighted in pink. Within the users, there is a variability in scores across different spaces. Users 3, 7, and 10 all have the highest scores within the spaces to which they have access. The other users have at least one metric with a better score in another space. Looking across all users, Answer Relevance scores are relatively high, above 0.8 in all cases, indicating that users generally provide answers that are on-topic and useful. However, Context Precision, Faithfulness, and Context Recall show greater variations. Faithfulness does not reach high levels, except for users 8 (0.5000) in space 1 and 9 (0.3906) in space 3. Context Recall also varies, with users achieving scores above 0.6 for some spaces, indicating a better coverage of relevant context in the answers.

Name	Space Key	Context Precision	Answer Relevance	Faithfulness	Context Recall
admin	**0**	0.4000	0.6990	**0.2990**	**0.6250**
	1	0.0000	**0.8232**	0.0545	0.2154
	2	0.0000	0.4630	0.1071	0.2000
	3	**0.6000**	0.3677	0.2000	0.4167
User 1	0	0.2000	0.6270	0.0714	0.4500
	1	0.6000	0.3517	0.1413	**0.7500**
	2	0.0000	**0.8554**	0.2222	0.2500
	3	**0.8000**	0.5132	**0.3343**	0.4500
User 2	0	0.0000	**0.8299**	**0.3126**	0.3000
	1	0.0000	0.4589	0.0000	**0.4250**
	2	0.0000	0.8170	0.0800	0.4000
	3	**0.2000**	0.8013	0.2167	0.4222
User 3	0	0.0000	0.6842	0.2822	0.4000
	1	0.0000	0.3282	0.2879	0.3000
	2	0.0000	0.6743	0.1938	0.4000
	3	**0.6000**	**0.8857**	**0.3504**	**0.5333**
User 4	**0**	**0.4000**	0.6615	**0.3355**	0.4000
	1	0.0000	0.4646	0.0000	0.2154
	2	0.0000	0.6552	0.0526	0.2000
	3	**0.4000**	**0.8058**	0.1836	**0.5327**
User 5	0	0.2000	0.6915	0.1709	0.4333
	1	**0.6000**	**0.8706**	**0.2500**	**0.7000**
	2	0.0000	0.8669	0.1125	0.2500
	3	**0.6000**	0.8070	0.2386	0.3333
User 6	0	0.0000	0.8042	0.0000	0.3000
	1	0.0000	0.8482	0.1810	**0.4250**
	2	0.0000	**0.8668**	0.0071	0.3000
	3	**0.2000**	0.8547	**0.3775**	0.3028
User 7	**0**	**0.4000**	**0.8885**	**0.3610**	0.4000
	1	0.0000	0.6456	0.0000	0.2000
	2	0.0000	0.8246	0.0222	0.2000
	3	**0.4000**	0.5201	0.0273	**0.6354**
User 8	0	0.2000	0.3333	0.1333	0.3800
	1	**0.6000**	0.6601	**0.5000**	**0.6500**
	2	0.0000	0.4412	0.2033	0.2500
	3	**0.6000**	**0.8148**	0.1500	0.1000
User 9	0	0.0000	**0.8669**	0.2170	0.4000
	1	0.0000	0.3385	0.1117	0.3000
	2	0.0000	0.6531	0.1800	0.2500
	3	**0.6000**	0.6484	**0.3906**	**0.6333**
User 10	**0**	0.4000	**0.8433**	**0.2606**	0.5000
	1	0.0000	0.6766	0.0000	0.2154
	2	0.0000	0.6462	0.1333	0.2500
	3	**0.6000**	0.6584	0.2333	**0.5333**

Table 6.6: The mean Context Precision, Answer Relevance, Faithfulness, and Context Recall scores for each user and the baseline. The accessible spaces for each user are highlighted in yellow, and the highest value for each metric per user is highlighted in blue. This detailed comparison illustrates how user-specific access impacts response quality across multiple evaluation criteria.

6.5 Manual Inspection

Considering the limitations of the above scores, a manual inspection is also conducted to assess response quality. Manual inspection enables an assessment of response quality that automated metrics alone may miss, such as subtle inaccuracies, coherence, or understanding of context. Besides evaluating the nuances in the responses, answers generated by clients should not include any sensitive or classified information to which they should not have access.

6.5.1 Responses with Permissions

During the inspection of the answers generated by different clients' local LLMs, notable differences emerged in generating responses, taking permissions into account. The answers in Table 6.7 to the first question already demonstrate permission enforcement (see Table 9.1 for the questions), where a client with access to the required documents to answer the question provides a concise, technical summary that focuses on essential information relevant to the question. This response is accurate and relevant, effectively capturing the information retrieved from the sensitive documents by addressing the key points outlined in the ablation study. In contrast, the other client generated a much longer response. This response resembles a more human-like thinking style and is not directly related to the specific research question.

Admin Response	**User 1 Response**
The ablation study in the R-GCN component demonstrates that the edge information is crucial for the model's performance. By removing nodes and edges, the study shows that the edge labels are significant for modeling. This indicates that the edge information is crucial for the model's performance, particularly in handling graph-based data where edge relationships are essential. The study's findings suggest that the edge information is crucial for the model's performance, and removing nodes and edges significantly affects the model's accuracy. This study helps us understand which parts of the graph are most important for the model's performance, providing insights into the graph's structure and the model's ability to handle different graph sizes.	Okay, so I'm trying to understand what the R-GCN component is and why it's significant within the Entity-GCN model. I'm not very familiar with graph neural networks (GCNs) or how they're used in this context, but I know a bit about machine learning and graph theory. Let me break this down step by step. First, what is a graph? A graph consists of nodes (also called vertices) and edges that connect these nodes. In the context of a study or research, the graph likely represents a set of entities with relationships between them. For example, in a social network, the nodes could be people and the edges could represent friendships. Now, what is an Entity-GCN? I think it's a type of graph neural network specifically designed to work with entities that have relationships. These entities could be people, places, organizations, or even other graphs. The idea is that each node in the graph is an entity, and the edges represent some kind of relationship or interaction between these entities. A paper I came across mentioned that the R-GCN (Relation Graph Convolutional Network) is a component of the Entity-GCN. The name "Relation" suggests that it's focused on relationships between entities rather than just any connections [...]

Table 6.7: Raw responses from admin and user 1, answering question 1 with space key 0 from Table 9.1: *"What is the significance of the R-GCN component within the Entity-GCN model based on the ablation study?"*. Admin has access to spaces 0 and 1, whereas user 1 has access to spaces 1 and 2. The response from user 1 is abbreviated with **[...]** for clarity.

6.5.2 Hallucinations

Hallucinations occur when a model produces information that is fabricated, misleading, or inconsistent with the data or known facts. RAG should mitigate the hallucinations by providing the information

needed to retrieve documents. Table 6.8 illustrates an example of hallucinations in responses. User 2 has no access to this document regarding *Bed Bath & Beyond* bankruptcy, so the local LLM started to hallucinate and states that "*The government of the United States filed for bankruptcy for Bed Bath & Beyond on January 27, 2018*", which is a statement that is incorrect and not supported by any real-world event. Meanwhile, the response from user 9 shows repetitive and nonsensical text, with parts of the phrase "The ruling was based on the case" repeated. The failure to generate factual and meaningful content results in an output that is meaningless and misleading. User 10 provides a more plausible answer by giving an answer directly to the question by stating that the company filed for bankruptcy. This response avoids fabricated information and does not show hallucinations.

User 2 Response	**User 9 Response**	**User 10 Response**
The government of the United States filed for bankruptcy for Bed Bath & Beyond on January 27, 2018. This action was taken under the U.S. Bankruptcy Act, specifically for the chapter on Industrial Property. The court ruled on the case, determining that the company had a valid bankruptcy claim and that it had a significant impact on the company's operations and provided a proposed bankruptcy settlement. The ruling was that the government could impose a $1.5 million settlement on the company's claims, but the specific terms of the settlement were not determined yet. [...] In summary, the government of the United States filed for bankruptcy for Bed Bath & Beyond on January 27, 2018, and provided a proposed bankruptcy plan and settlement. The government's action was to address potential financial obligations and to consider further legal steps if needed. The case is ongoing and there is no final ruling yet.	Okay, so I need to figure out if the Bed Bath & Beyond company filed for bankruptcy. Let me start by recalling what I know about bankruptcy. First, bankruptcy is a legal process where a company stops operating its business and is legally taken over. [...] The ruling was based on the case of Bed Bath & Beyond, which had the case against the company, and the court ruled in favor of the company. The ruling was based on the case of Bed Bath & Beyond, which had the case of the company and its operations. I should also note that the ruling was based on the case of the case of Bed Bath & Beyond, which had the case of the company and its case. **The ruling was based on the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of the case of** [...]	**The company Bed Bath & Beyond filed for bankruptcy** because they couldn't liquidate their own assets and pay off their debts. The exact details of the bankruptcy process, including the amount of money they had and the court process, would need to be looked up for more information. In summary, Bed Bath & Beyond filed for bankruptcy because they couldn't liquidate their own assets and pay off their debts, and they sold their own shares to cover their debts.

Table 6.8: Raw responses from user 2 and user 9, answering question 12 with space key 3 from Table 9.1: *"Did Bed Bath & Beyond file for bankruptcy?"*. User 2 has access to space 2, whereas both users 9 and 10 have access to space 3. Highlighted in yellow are the hallucination-generated texts from users 2 and 9. The blue highlight marks a correct and concrete answer to the question. Responses from users 2 and 9 are abbreviated with [...] for clarity.

Chapter 7

Discussion

This chapter discusses the various results obtained during the evaluation of our framework. In the first section, the key scores and metrics will be interpreted. The following sections will contain the challenges during both the implementation and practical use of the framework.

7.1 Interpretation of the Results

In this section, the IXN Scores, RAGAS metrics, ROUGE, and BLUE metrics will be further interpreted based on the results in chapter 6. Besides interpreting the results, these metrics will also be discussed based on their limitations.

7.1.1 IXN Scores

Further interpreting the IXN results, the results were inconclusive in fully validating its success. Question 14 for users 3 and 9 and question 16 for users 2 and 6 have the highest IXN scores, which are 0.6. These results are noteworthy because they suggest that 60% of the top papers that the secure system retrieved were the same as those that the baseline system retrieved. Specifically, users 2, 3, 6, and 9 had the highest average IXN score of 0.22, indicating that their retrievals overlapped most closely with the baseline. According to the definition of the IXN score, these users should have the most access to documents across permission boundaries. However, this contradicts the actual permissions these users have, as shown in Table 5.3, where users 3 and 9 only have access to one space each. This could indicate a concentration of relevant documents in those spaces. On the other hand, users with limited access, such as 4 and 7, have lower IXN scores, indicating that access control prevented them from retrieving many documents that the baseline would have returned.

As suggested by Chen [53], the IXN score was used to evaluate retrieval security. While the results from Chen were significant, the results from this research deviate notably. The key difference here lies in the implementation; in our framework, a separate vector store is instantiated for each client, including the baseline. This decision results in variations in the retrieval of documents for semantically similar queries. While there are cases where clients do have some overlapping documents, this overlap is not consistent across all queries (as shown in Table 6.2). This variability highlights the sensitivity of this metric to the storage and configuration of vectors for clients. In Chen's case, retrieval is performed on only a few vector stores, rather than creating a separate vector store for each client.

During the implementation phase, experiments have been conducted using a single vector store for document retrieval. Using a shared vector store is a straightforward method since it enables more

consistent and higher-quality retrieval, as all clients benefit from a larger and unified knowledge base. However, this would be contrary to the security principles of using this framework in an environment that handles classified documents. Sharing embeddings in a global vector store, even if noise or anonymization is added to them, introduces a potential risk as the vectors become vulnerable to inference attacks, embedding inversion, or membership inference, where an adversary could potentially extract sensitive information from the shared vector store.

7.1.2 ROUGE and BLEU

The analysis of response quality reveals nuanced variations in how document access policies influence the effectiveness of RAG across different users. BLEU, ROUGE-1, and ROUGE-L provide insight into the fluency, precision, and recall of generated answers. Across nearly all users, the highest BLEU and ROUGE scores were found within the spaces to which they had explicit access. For example, user 3 performed best in space key 3, which they had access to, and similarly for users 1, 5, 6, and 10. This supports the hypothesis that document access policies have a direct impact on the quality of RAG outputs. Users with access to only a subset of documents underperformed for the recall metrics. This suggests that their generated answers were limited by insufficient context, which highlights the effect of restricted access on the quality of question answering. Some users generated short but precise answers, characterized by high precision and low recall. In contrast, others offered a more extensive but less focused response, marked by high recall but lower precision or BLEU score.

The dataset DOCBENCH includes sufficient variety across space keys to demonstrate the nuanced effects of document access. Precision and recall varied significantly across users, which confirms that the dataset was complex and diverse enough to expose real differences. When interpreting response quality based on the ROUGE and BLEU scores, we note that these metrics are computed using exact word matching rather than semantic content, which is a key concern in RAG. These metrics are used widely due to their simplicity and computational efficiency, but they have notable limitations. This limitation rises especially in the context of RAG, where responses could have semantically equivalent meanings, yet exact word matching would consider these answers different. The retrieved content may be phrased differently from the true answers, leading to a low overlap in exact word matching, even if the generated response is accurate and relevant. While this delivered a measure of precision, it did not fully reflect the architecture's ability to generate semantically accurate and relevant responses. Therefore, relying only on exact word matching could lead to an underestimation of the model's true performance.

Additionally, the results represent an average across all questions, including those for which users did not have access to the necessary documents to answer them accurately. Questions for which users do not have access lead to lower scores, which reduces overall precision, recall, BLEU, and ROUGE scores when taking the average.

7.1.3 RAGAS metric

The RAGAS metrics were used to evaluate response quality, including the following metrics: Context Precision, Answer Relevance, Faithfulness, and Context Recall. Unlike ROUGE and BLEU, evaluating with this method gives more insight into the semantic content of the generated answers by the local LLMs of the clients. As the results from Table 6.5 do indicate the influence of implementing FL with security measures and permission enforcement, Table 6.6 gives more insight into how the RAGAS metrics vary when compared per space key instead of per question. For most users, the majority of the highest scores in pink correspond to the spaces to which they have access. Admin has, for example, a high Context Precision for space 3, while it only has access to spaces 0 and 1. The user where this significantly stands out is user 6, where only Answer Relevance is the highest for space

2. After manually inspecting some of the generated answers by user 6, and given that user 6 was only trained for one communication round (see Table 6.1), we can conclude that this client was not sufficiently trained to perform accurately within the spaces to which they had access. Other users perform exactly as expected according to the accessible spaces, such as users 3, 5, 7, and 10.

Using `llama3:8b` as a judge LLM, a consistent evaluation was performed to compute the RAGAS scores. This model was selected for its robust instruction-following capabilities and balanced performance in comprehension and factual judgment tasks. Using such a capable judge model helps approximate human-level evaluations while allowing reproducible assessments across a wide set of client answers. The results for RAGAS suggest that for accurate outcomes, high-quality reasoning and effective retrieval methods are essential for generating grounded responses from federated RAG systems.

7.2 Challenges in Federated Learning

Implementing FL for LLMs introduces several complexities. Maintaining privacy across clients while allowing shared updates is one of these hurdles. One of the main questions about why these shared weights can be shared safely is due to the implementation of DP and HE. The proposed method defends against inference attacks as the server can only access encrypted LoRA parameters. After aggregation, only updates are shared with the clients, which are then loaded into the local model.

FL training is slower to converge compared to centralized learning due to sparse or even conflicting updates from clients, as is shown in Figure 6.1. When large LLMs are used for FL, communication bottlenecks also arise. LoRA helps mitigate this by reducing the sizes of the updates, but the overall communication round complexity remains high, especially if this setup is deployed in the real world. Training this setup with the aforementioned parameters took 30 to 50 minutes, which is manageable for a training setup, but not for a real-world case environment where a larger LLM is preferable.

Another challenge is ensuring fairness among clients. Some clients may drastically vary in terms of data quantity and document access. Without careful aggregation, clients with more data may dominate model learning, resulting in even greater communication overhead. This could also lead to a biased performance that benefits only a specific group of users with the same role. Ensuring that each user benefits equally from FL training remains an aspect to be researched within the FL domain.

Finally, FL systems are still vulnerable to a variety of threats. A thorough security assessment is still required, even though this thesis presents mitigation techniques like HE and DP. In particular, inference attacks, such as prompt injections, were not addressed in this research and were outside of the current scope.

7.3 Challenges of using DeepSeek

One of the challenges of using the smallest model of DeepSeek, `DeepSeek-R1-Distill-Qwen-1.5B`, is that the rate of hallucinations is high, often producing outputs that are factually incorrect or structurally incoherent. This issue is apparent in several generated responses, where not only fabricated claims but also repetitive words were present, significantly diminishing the readability and credibility of the answers, as shown in Table 6.8. Client 9's model frequently generated text with repetitive words. These hallucinations can undermine the purpose of safely using an LLM on classified data, as it may produce incorrect or misleading information. In high privacy-sensitive environments, the risk of hallucinations can be as problematic as direct unauthorized access.

Furthermore, the presence of hallucinations complicates evaluation. BLEU and ROUGE fail to penalize hallucinated content if it accidentally lexically overlaps with the ground truth. RAGAS

metrics may not fully capture the nuances of these repetitions and may also penalize them. This also highlights a broader limitation in current LLM evaluation, where evaluating an LLM without human evaluators remains a significant challenge.

7.4 Real-World Implications

Due to significant computational demands from the FL training setup, the experiments are limited to eleven clients. This decision was primarily due to the available hardware resources, as training with local LLMs requires substantial RAM and processing power. Each client hosts a local LLM, managing its retrieval and participating in the FL process, all of which do not scale in practice with the limited hardware. Additionally, HE effectively preserves privacy, but it introduces additional computational and memory overhead. These same limitations ensured that only the smallest DeepSeek model could effectively be run across all clients without using too much of the hardware's memory capacity. Larger variants of DeepSeek were ruled out due to the extensive memory usage. The findings and scalability of the setup should be interpreted in light of these computational constraints.

Chapter 8

Conclusion

This research explored the integration of FL with LLMs in the context of RAG, with a strong focus on secure and privacy-preserving model training across distributed environments that contain sensitive data. A practical framework was implemented using PEFT, specifically LoRA, local RAG, and privacy methods such as HE and DP. The findings from this work are summarized below, organized by the primary research questions.

How can federated learning be used to train LLMs on sensitive data without exposing that data?

This research demonstrated that FL can be adapted to enable training LLMs across clients that hold sensitive documents. By fine-tuning the local models of the clients with LoRA, clients went through a training process while maintaining complete control over their private documents, which were decentralized. The LoRA weights enabled aggregation at the server side in the setup, thereby reducing communication overhead. Each client managed the retrieval and inference of their local LLM. HE and DP were implemented to protect the weights during communication and aggregation. This framework preserved data privacy using these mechanisms while still allowing for meaningful global model improvements across communication rounds.

What are the risks of using LLMs in a federated learning setup without taking security into account, and what mitigation can be used?

FL systems are vulnerable to a range of adversarial threats, such as eavesdropping on model weights, gradient leakage, and poisoning attacks. Without mitigations, a central server could exploit parameter updates, a malicious client could poison the dataset, or an adversary could steal the weights. To counter this, HE was used to protect updates from being inspected by a curious server, and DP added noise to further obscure sensitive gradients. Although these methods introduced computational difficulties, the security guarantees they provide are essential to call this proposed framework safe and privacy-aware.

How does permission enforcement affect the accuracy of LLM responses for authorized clients?

Clients receive answers based only on the documents they are permitted to access. This thesis introduced a permission-aware retrieval and inference pipeline that enforced document-level access control (also see section 4.4). While it was, for most users, effective in preventing data leakage,

permission enforcement also introduced challenges. Limiting the context window to only the client's accessible documents sometimes resulted in lower BLEU, ROUGE, and RAGAS scores. However, according to the different metrics used, the framework restricted correctly information for some of the users, for admin, user 1, 3, 5, 6, 7, and 10 for BLEU/RAGAS (see Table 6.4) and for users 3, 7, 10 for RAGAS (see Table 6.6). The evaluation also showed that users with fewer permissions sometimes received responses with lower semantic relevance, which highlights a trade-off: stronger security may reduce answer accuracy.

This research presents a practical approach to using FL for LLMs with document-level security policies. Secure, privacy-aware, and FL training is feasible without requiring centralization of the users' data. The framework achieves this by combining secure model update mechanisms, LoRA, HE, and DP, and decentralized RAG modules with permission enforcement. The findings highlight the trade-offs between access control and model performance. As more security measurements are required, the semantics and accuracy of answers may degrade. However, in sensitive domains, such trade-offs are necessary to comply with the privacy and security requirements.

8.1 Future Work

While this research has demonstrated the applicability of using FL for LLMs and incorporating privacy measures to enhance security, several directions remain open for future work to further improve both the framework's robustness and its practical applicability. A critical next step is to use more clients to make it more realistic, resembling the real world. Extending the framework with more clients, possibly in the hundreds, would better simulate production environments such as healthcare, governmental setups, or large enterprises. Achieving this scale will require optimizing the memory usage of client models, such as parallelizing computations more effectively. Model diversity is another challenge for exploration. In this research, all clients used the same base model. However, real-world scenarios may involve different LLMs with additional capabilities or specializations. Evaluating how these different models affect convergence and response quality in FL still remains unexplored.

In terms of privacy and security, the framework utilizes HE and DP to ensure the safe deployment of model updates. Future efforts could potentially concentrate on applying a different technique called pruning. Before sending the clients to the server, some of the weights can be pruned to ensure both efficiency and security. Less important weights can be set to zero after PEFT, focusing computational resources on the most relevant parameters.

As previously discussed, a persistent challenge is the evaluation strategy, which reflects an unresolved trade-off in current research between efficiency and semantic accuracy. This research utilized automated metrics, such as RAGAS, ROUGE, and BLEU, which fall short in either capturing semantics or user relevance. Future research could incorporate more sophisticated evaluation methods, including those using semantic similarity models or human-in-the-loop systems.

Bibliography

[1] A. Toosi, A. G. Bottino, B. Saboury, E. Siegel, and A. Rahmim, "A brief history of ai: How to prevent another winter (a critical review)," *PET clinics*, vol. 16, no. 4, pp. 449–469, 2021.

[2] N. Muthukrishnan, F. Maleki, K. Ovens, C. Reinhold, B. Forghani, and R. Forghani, "Brief history of artificial intelligence," *Neuroimaging Clinics*, vol. 30, no. 4, pp. 393–399, 2020.

[3] J. T. Force, "Security and privacy controls for information systems and organizations," National Institute of Standards and Technology, Tech. Rep., 2017.

[4] M. R. Na and K. Sundharakumar, "A study on air-gap networks," in *2024 5th International Conference on Innovative Trends in Information Technology (ICITIIT)*, IEEE, 2024, pp. 1–6.

[5] B. Cottier, R. Rahman, L. Fattorini, N. Maslej, T. Besiroglu, and D. Owen, "The rising costs of training frontier ai models," *arXiv preprint arXiv:2405.21015*, 2024.

[6] 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.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[8] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.

[9] H. U. Sami, S. Sen, A. K. Roy-Chowdhury, S. V. Krishnamurthy, and B. Guler, "Gradient inversion attacks on parameter-efficient fine-tuning," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 10 224–10 234.

[10] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE access*, vol. 10, pp. 22 359–22 380, 2022.

[11] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "{Batchcrypt}: Efficient homomorphic encryption for {cross-silo} federated learning," in *2020 USENIX annual technical conference (USENIX ATC 20)*, 2020, pp. 493–506.

[12] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," *Future Generation Computer Systems*, vol. 150, pp. 272–293, 2024.

[13] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106 775, 2021.

[14] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[15] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, T. Yu, G. Wang, and Y. Chen, "Towards building the federatedgpt: Federated instruction tuning," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 6915–6919.

[16] C. Chen, X. Feng, Y. Li, L. Lyu, J. Zhou, X. Zheng, and J. Yin, "Integration of large language models and federated learning," *Patterns*, vol. 5, no. 12, 2024.

[17] G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, *et al.*, "Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions," *IEEE Access*, 2024.

[18] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[19] Z. Wang, Z. Chu, T. V. Doan, S. Ni, M. Yang, and W. Zhang, "History, development, and principles of large language models: An introductory survey," *AI and Ethics*, pp. 1–17, 2024.

[20] F. Liu, Y. Liu, L. Shi, H. Huang, R. Wang, Z. Yang, L. Zhang, Z. Li, and Y. Ma, "Exploring and evaluating hallucinations in llm-powered code generation," *arXiv preprint arXiv:2404.00971*, 2024.

[21] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[22] C. Ren, H. Yu, H. Peng, X. Tang, B. Zhao, L. Yi, A. Z. Tan, Y. Gao, A. Li, X. Li, *et al.*, "Advances and open challenges in federated foundation models," *arXiv preprint arXiv:2404.15381*, 2024.

[23] 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," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5260–5271.

[24] J. Jiang, H. Jiang, Y. Ma, X. Liu, and C. Fan, "Low-parameter federated learning with large language models," in *International Conference on Web Information Systems and Applications*, Springer, 2024, pp. 319–330.

[25] Y. Sun, Z. Li, Y. Li, and B. Ding, "Improving lora in privacy-preserving federated learning," *arXiv preprint arXiv:2403.12313*, 2024.

[26] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[27] Y. Shi, O. Kotevska, V. Reshniak, A. Singh, and R. Raskar, "Dealing doubt: Unveiling threat models in gradient inversion attacks under federated learning, a survey and taxonomy," *arXiv preprint arXiv:2405.10376*, 2024.

[28] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[29] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, *et al.*, "Opacus: User-friendly differential privacy library in pytorch," *arXiv preprint arXiv:2109.12298*, 2021.

[30] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[31] V. Pihur, A. Korolova, F. Liu, S. Sankuratripati, M. Yung, D. Huang, and R. Zeng, "Differentially-private" draw and discard" machine learning," *arXiv preprint arXiv:1807.04369*, 2018.

[32] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," *arXiv preprint arXiv:2009.03561*, 2020.

[33] W. Wei, L. Liu, Y. Wu, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2021, pp. 797–807.

[34] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[35] D. Reis, J. Takeshita, T. Jung, M. Niemier, and X. S. Hu, "Computing-in-memory for performance and energy-efficient homomorphic encryption," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2300–2313, 2020.

[36] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame, *et al.*, "Safelearn: Secure aggregation for private federated learning," in *2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021, pp. 56–62.

[37] J. So, C. He, C.-S. Yang, S. Li, Q. Yu, R. E Ali, B. Guler, and S. Avestimehr, "Lightsecagg: A lightweight and versatile design for secure aggregation in federated learning," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 694–720, 2022.

[38] H. Zhu, R. S. M. Goh, and W.-K. Ng, "Privacy-preserving weighted federated learning within the secret sharing framework," *IEEE Access*, vol. 8, pp. 198 275–198 284, 2020.

[39] M. Mansouri, M. Önen, W. B. Jaballah, and M. Conti, "Sok: Secure aggregation based on cryptographic schemes for federated learning," *Proceedings on Privacy Enhancing Technologies*, 2023.

[40] P. Subramaniam and S. Krishnan, "Intent-based access control: Using llms to intelligently manage access control," *arXiv preprint arXiv:2402.07332*, 2024.

[41] R. Sandhu and Q. Munawer, "How to do discretionary access control using roles," in *Proceedings of the third ACM workshop on Role-based access control*, 1998, pp. 47–54.

[42] Y. Fan, Z. Han, J. Liu, and Y. Zhao, "A mandatory access control model with enhanced flexibility," in *2009 international conference on multimedia information networking and security*, IEEE, vol. 1, 2009, pp. 120–124.

[43] R. Kuhn, E. Coyne, and T. Weil, "Adding attributes to role-based access control," 2010.

[44] S. Kohler, *Atlassian confluence 5 essentials*. Packt Publishing Ltd, 2013.

[45] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "Fate-llm: A industrial grade federated learning framework for large language models," *arXiv preprint arXiv:2310.10049*, 2023.

[46] Z. Zhang, J. Zhang, J. Huang, L. Qu, H. Zhang, Q. Wang, X. Zhou, and Z. Xu, "Fewfedpit: Towards privacy-preserving and few-shot federated instruction tuning," *arXiv preprint arXiv:2403.06131*, 2024.

[47] N. Kokash, L. Wang, T. H. Gillespie, A. Belloum, P. Grosso, S. Quinney, L. Li, and B. de Bono, "Ontology-and llm-based data harmonization for federated learning in healthcare," *arXiv preprint arXiv:2505.20020*, 2025.

[48] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[49] E. Jiang, A. Chen, I. Tenison, and L. Kagal, "Medirag: Secure question answering for healthcare data," in *2024 IEEE International Conference on Big Data (BigData)*, IEEE, 2024, pp. 6476–6485.

[50] D. Wang and S. Zhang, "Large language models in medical and healthcare fields: Applications, advances, and challenges," *Artificial intelligence review*, vol. 57, no. 11, p. 299, 2024.

[51] C. Zakka, R. Shad, A. Chaurasia, A. R. Dalal, J. L. Kim, M. Moor, R. Fong, C. Phillips, K. Alexander, E. Ashley, *et al.*, "Almanac—retrieval-augmented language models for clinical medicine," *Nejm ai*, vol. 1, no. 2, AIoa2300068, 2024.

[52] Z. Chen, A. H. Cano, A. Romanou, A. Bonnet, K. Matoba, F. Salvi, M. Pagliardini, S. Fan, A. Köpf, A. Mohtashami, *et al.*, "Meditron-70b: Scaling medical pretraining for large language models," *arXiv preprint arXiv:2311.16079*, 2023.

[53] A. Chen, "Policy-based access control in federated clinical question answering," Ph.D. dissertation, Massachusetts Institute of Technology, 2024.

[54] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai, *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[55] L. Mondrejevski, I. Miliou, A. Montanino, D. Pitts, J. Hollmén, and P. Papapetrou, "Flicu: A federated learning workflow for intensive care unit mortality prediction," in *2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, 2022, pp. 32–37.

[56] M. Pennisi, F. P. Salanitri, G. Bellitto, B. Casella, M. Aldinucci, S. Palazzo, and C. Spampinato, "Feder: Federated learning through experience replay and privacy-preserving data synthesis," *Computer Vision and Image Understanding*, vol. 238, p. 103 882, 2024.

[57] Z. L. Teo, L. Jin, N. Liu, S. Li, D. Miao, X. Zhang, W. Y. Ng, T. F. Tan, D. M. Lee, K. J. Chua, *et al.*, "Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture," *Cell Reports Medicine*, vol. 5, no. 2, 2024.

[58] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106 854, 2020.

[59] Y. Gao, L. Liu, B. Hu, T. Lei, and H. Ma, "Federated region-learning for environment sensing in edge computing system," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2192–2204, 2020.

[60] A. Abu Jabal, E. Bertino, J. Lobo, D. Verma, S. Calo, and A. Russo, "Flap-a federated learning framework for attribute-based access control policies," in *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, 2023, pp. 263–272.

[61] A. P. Kalapaaking, I. Khalil, and M. Atiquzzaman, "Smart policy control for securing federated learning management system," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1600–1611, 2023.

[62] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of healthcare informatics research*, vol. 5, no. 1, pp. 1–19, 2021.

[63] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[64] W.-S. Choi, B. Reagen, G.-Y. Wei, and D. Brooks, "Impala: Low-latency, communication-efficient private deep learning inference," *arXiv preprint arXiv:2205.06437*, 2022.

[65] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," in *Proceedings of the 55th annual design automation conference*, 2018, pp. 1–6.

[66] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEe Access*, vol. 9, pp. 63 229–63 249, 2021.

[67] T. V. Carstens, E. Ebrahimi, G. N. Tabia, and D. Unruh, "Relationships between quantum ind-cpa notions," in *theory of cryptography conference*, Springer, 2021, pp. 240–272.

[68] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[69] J. Hunker and C. W. Probst, "Insiders and insider threats-an overview of definitions and mitigation techniques.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 2, no. 1, pp. 4–27, 2011.

[70] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25*, Springer, 2020, pp. 480–501.

[71] J. Zhang, H. Zhu, F. Wang, J. Zhao, Q. Xu, and H. Li, "Security and privacy threats to federated learning: Issues, methods, and challenges," *Security and Communication Networks*, vol. 2022, no. 1, p. 2 886 795, 2022.

[72] E. Evstafev, "Token-hungry, yet precise: Deepseek r1 highlights the need for multi-step reasoning over speed in math," *arXiv preprint arXiv:2501.18576*, 2025.

[73] T. Zhuang, C. Kuang, X. Li, Y. Teng, J. Wu, Y. Wang, and L. Shang, "Docpuzzle: A process-aware benchmark for evaluating realistic long-context reasoning capabilities," *arXiv preprint arXiv:2502.17807*, 2025.

[74] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[75] S. Halevi and V. Shoup, "Algorithms in helib," in *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, Springer, 2014, pp. 554–571.

[76] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity text embeddings through Self-Knowledge distillation," *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2318–2335, Jan. 2024. DOI: 10.18653/v1/2024.findings-acl.137. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.137.

[77] A. Zou, W. Yu, H. Zhang, K. Ma, D. Cai, Z. Zhang, H. Zhao, and D. Yu, "Docbench: A benchmark for evaluating llm-based document reading systems," *arXiv preprint arXiv:2407.10701*, 2024.

[78] Ttorvela, *Space-permissions-handler*, https://github.com/ttorvela/space-permissions-handler, Accessed June 2025.

[79] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[80] Y. B. Wiryen, N. A. Vigny, M. J. Ngono, and F. L. Aimé, "Leveraging tenseal: A comparative study of bfv and ckks schemes for training ml models on encrypted iot data," *International Journal of Information Security and Privacy (IJISP)*, vol. 18, no. 1, pp. 1–17, 2024.

[81] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[82] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[83] S. Es, J. James, L. E. Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2024, pp. 150–158.

[84] *Context precision*, `https://docs.ragas.io/en/latest/concepts/metrics/available_metrics/context_precision/`, Accessed June 2025.

[85] *Answer relevancy*, `https://docs.ragas.io/en/v0.1.21/concepts/metrics/answer_relevance.html`, Accessed June 2025.

[86] *Faithfulness*, `https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/faithfulness/`, Accessed June 2025.

[87] *Context recall*, `https://docs.ragas.io/en/latest/concepts/metrics/available_metrics/context_recall/`, Accessed June 2025.

Chapter 9

Appendix

9.1 Prompt Template

Listing 9.1: Prompt template used for generation

```
messages = [
        {
            "role": "system",
            "content": """
            You are an expert assistant that provides only direct, verified answers.

            Rules:
            1. Never show your reasoning, thought process, or analysis steps.
            2. Answer immediately with the final, most accurate response.
            3. If unsure, say: "I don't have verified information on this."
            4. Avoid all filler language, including:
            - "Okay, so I need to figure out..."
            - "Let me break this down..."
            - "I'm not very familiar, but..."
            5. For lists or categories, state them without introduction.
            6. If the question requires a source (e.g., reports), cite it or admit uncertainty.
            """
        },
        {
            "role": "user",
            "content": f"""
            Context (optional):
            {context}

            Question:
            {query}
            """
        }
    ]
```

9.2 Questions

#	Space Key	Question
1	0	What is the significance of the R-GCN component within the Entity-GCN model based on the ablation study?
2	0	What are the consequences for a foreign national who changes their resident status and moves out of Iran without transferring their real estate rights?
3	0	Is the IBGE allowed to share data with public or private companies for generating official statistics?
4	0	What model does China Exim Bank follow, which allows the use of natural resources as collateral for loans?
5	0	Has the United Kingdom introduced a bill concerning the regulation of AI and workers' rights?
6	1	Which training signal resulted in the highest score for the CITE category?
7	1	Which model showed the largest improvement in AUC value after the addition of DSGAN?
8	1	What is the Word Error Rate (WER) of the ATD spell-corrector model for the 'Key' attack?
9	1	What might hinder the U.S.'s ability to achieve strategic prevention of religious freedom violations?
10	1	What is the energy efficiency target for Toyota's compact SUV in the bZ series?
11	2	How does the performance of LSTUR-con measured by AUC compare to LSTUR-ini as the mask probability p increases?
12	2	What are the four distinct focus areas of Novo Nordisk's corporate strategy?
13	2	Did Bed Bath & Beyond file for bankruptcy?
14	2	What accomplishments had Louise Glück achieved before receiving the Nobel Prize?
15	2	What positions did Dr. Reese hold at the Company between 2015 and 2018?
16	3	What is the BLEU score for CodeBERT when pre-trained with MLM+RTD objectives?
17	3	Which metric showed a statistically significant increase in correlation with human assessment over all other metrics for the German to French language pair according to Williams test?
18	3	What is the performance score for Entity Recognition when multitasked with Coreference Resolution?
19	3	Why are the results of BERT with a mean of 71.6% not considered meaningful?
20	3	On which platform were the annotations for the BoolQ dataset collected?

Table 9.1: List of questions categorized by their corresponding space key, providing an overview of the topics and their associated identifiers used in the study.

9.3 Extended Evaluation Metrics

ID	BLEU	ROUGE1-P	ROUGE1-R	ROUGE1-F	ROUGEL-P	ROUGEL-R	ROUGEL-F
0	0.1131	0.1885	0.4600	0.2674	0.1393	0.3400	0.1977
1	0.0352	0.1195	0.3878	0.1827	0.1006	0.3265	0.1538
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0245	0.0012	0.5000	0.0024	0.0012	0.5000	0.0024
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.1589	0.1064	0.6250	0.1818	0.1064	0.6250	0.1818
6	0.0246	0.1186	0.6364	0.2000	0.1102	0.5909	0.1857
7	0.0456	0.0547	0.6111	0.1005	0.0547	0.6111	0.1005
8	0.0313	0.0026	0.3333	0.0051	0.0026	0.3333	0.0051
9	0.0000	0.0048	0.6000	0.0095	0.0032	0.4000	0.0063
10	0.1196	0.1282	0.5172	0.2055	0.1111	0.4483	0.1781
11	0.0130	0.0201	0.5000	0.0387	0.0147	0.3636	0.0282
12	0.0403	0.1429	1.0000	0.2500	0.1429	1.0000	0.2500
13	0.0216	0.0058	0.6000	0.0114	0.0058	0.6000	0.0114
14	0.0375	0.0098	0.5000	0.0193	0.0074	0.3750	0.0145
15	0.1555	0.0177	0.7333	0.0345	0.0177	0.7333	0.0345
16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0137	0.0321	0.2727	0.0575	0.0286	0.2424	0.0511
19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 9.2: Response quality metrics for Client 1.

Question	Context Precision	Answer Relevance	Faithfulness	Context Recall
1	1.00	0.84	0.40	1.00
2	1.00	0.92	1.00	1.00
3	0.00	0.00	0.00	nan
4	0.00	0.87	0.00	0.50
5	0.00	0.87	0.10	0.00
6	0.00	0.76	0.00	1.00
7	0.00	0.86	0.00	0.00
8	0.00	0.81	0.27	0.00
9	0.00	0.81	0.00	0.00
10	0.00	0.88	0.00	0.08
11	0.00	0.63	0.00	1.00
12	0.00	0.00	nan	0.00
13	0.00	0.89	0.00	0.00
14	0.00	0.80	0.43	0.00
15	0.00	0.00	0.00	0.00
16	0.00	0.93	0.00	0.00
17	0.00	0.91	0.00	nan
18	1.00	0.00	0.00	0.00
19	1.00	0.00	1.00	1.00
20	1.00	0.00	0.00	0.67

Table 9.3: Response quality score for Client 1.

9.4 Code Repository

The complete source code for the framework developed in this thesis is publicly available on GitHub: `https://github.com/pcrooijendijk/Master_Thesis.git`
The repository includes the following components:

- The full implementation of the framework discussed in this thesis.
- Datasets used for testing, evaluation, and training.
- Scripts for data preprocessing, model training, and other essential functions to run the framework.

For instructions on how to set up and use the code, please refer to the `README.md` file included in the repository.