# Addressing IP Protection in RAG Systems by Text Obfuscation

Final Year Project

May 1, 2025

Jiang Zhejun 3035855868

Supervisor: Prof. Chow Ka-Ho

Department of Computer Science, HKU

## Abstract

Large language models (LLMs) like GPT-3.5, GPT-4, and Llama 3 excel in various tasks but face limitations, including outdated knowledge, lack of domain specificity, and error-prone outputs [3, 18, 21]. To enhance their capabilities, Retrieval-Augmented Generation (RAG) [16] systems incorporate external knowledge databases, addressing some of these challenges. However, RAG systems pose significant risks related to privacy breaches and intellectual property infringement due to their reliance on web-crawled data. This study explores methods to mitigate these risks by proposing a strategy to generate texts that RAG systems cannot query or utilize, thereby safeguarding sensitive information. Two innovative approaches to enhance protection against potential breaches of intellectual property and privacy are investigated: (1) **Retrieval Stage**: Inspired by unlearnable examples [20], to avoid the private data from being retrieved by retriever, an optimization is used to modify the original text to minimize similarity scores between the private document and related queries; while at the same time, a shadow document with misinformation or non-private data is created which aims to maximize the similarity score and dominate the retrieving result; and (2) **Generation Stage**, By uploading watermarked private data and applying watermark-triggered attack towards the LLM, false or no information can generated by the RAG system. These methods are then tailored for different attacker knowledge scenarios—black-box and white-box settings. Preliminary experiments showed that compared to baseline settings, after applying the methods proposed in this work, there are significant improvements in IP Protection. Lastly, the impractical assumptions of attack towards generation stage attack and possible future improvements are discussed.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Modern large language models (LLMs) have demonstrated remarkable performance in various tasks, including question answering, reading comprehension, text summarization, mathematical reasoning [21]. Models like GPT-3.5 [1], GPT-4 [3], and Llama 3 [18] are widely used for their generative capabilities. However, these models are still constrained in the following ways: (1) these models have limited scope of knowledge and lack the most up-to-date information due to static pre-training data with a cutoff date; (2) they lack domain-specific knowledge unless fine-tuned; and (3) they still suffer from errors or hallucinations when tackling more complex or time-sensitive tasks, generating plausible but inaccurate text. These limitations pose challenges for deploying LLMs effectively in fields such as healthcare, finance, law, and scientific research. Addressing these issues is crucial for enhancing their applicability and reliability.

To address these limitations, Retrieval-Augmented Generation (RAG) [16] systems consisting knowledge database, retriever, and LLM augment the LLM generation with an external source of knowledge to be retrieved from a knowledge database. The knowledge database contains a large number of texts from various sources including Wikipedia, news articles, social media, and online community. A retriever retrieves the texts mostly related to the user's query from the knowledge database. Then, the texts are used as context to augment the generation and reduce hallucination by allowing LLMs to gain context knowledge.

To provide better services, companies like OpenAI use web crawlers [12] to routinely crawl texts from the Internet and use them in different stages including pre-training, fine-tuning, and the building of knowledge database for RAG. This can cause risks of infringement of intellectual property and privacy. Researchers have been working towards making private data unlearnable [20] [6] [11], but little progress has been done on preventing RAG systems from querying and generating using private data.

Empirical research indicates that RAG systems are susceptible to leaking their private retrieval databases [15], raising concerns about potential privacy breaches. Apart from that, RAG may further bring concerns in infringement of intellectual properties if the knowledge database contains copyrighted materials [14].

In the light of building ethical and responsible AI systems, this work explores means to resolve the issue of privacy leaks and intellectual property infringement caused by private data being queried and used for generation in RAG systems. Extending the idea of making private data unlearnable [20] [6], this work proposes a novel approach to generate texts that cannot be utilized by RAG systems. Depending on the background knowledge (e.g., black-box and white-box settings) of an attacker on RAG systems, two solutions will be presented to solve the text obfuscation problem, respectively. Extensive evaluation will be carried out, along with comparison to baseline settings.

The remainder of this report proceeds as follows. In Section 2, work related to IP Protection for RAG Systems are presented, including recently popular methods in attacking RAG systems and making examples unlearnable; in Section 3, the primary objective of the project and related assumptions are explained; in Section 4, the methodology employed is introduced; in Section 5, experiments are carried out and results are presented, along with comparison to other baselines, and ablation studies and effects against defenses are studied; in Section 6, the conclusions and discussions of this work are presented; in Section 7, the schedule of the project is presented.

Note that the experiments and conclusions are still in early stages.

## 2 Background and Related Work

In this section, theoretical background and related works that are essential in understanding the remainder of this work are reviewed. Section 2.1 offers a brief overview of the working principles of Retrieval-Augmented Generation (RAG) systems; Section 2.2 discusses current methods for protecting private data from deep learning models, noting that these methods do not apply in the RAG context, leaving a research gap which is the focus of this work; Section 2.3 examines available attack methods targeting RAG systems, emphasizing that these approaches stem from an adversarial perspective and pursue different objectives than our focus on data preservation. Despite that, due to the behaviors of attacked RAG systems, some attacking methods in the adversarial setting can be tailored and used in the our new data protection setting; Section 2.4 introduces the concept of text watermarking, a technique that will be utilized in the subsequent section to protect private data during the generation stage of RAG systems.

## 2.1 Retrieval Augmented Generation (RAG)

RAG [16] is a technique used to ground the generation from an LLM to a related textual corpus from a knowledge database to provide domain context, minimize hallucinations, and ensuring data freshness without requiring expensive fine-tuning or re-training operations. As shown in Figure 1, a RAG system typically includes three main components: a knowledge database, a RAG retriever and an LLM generator.

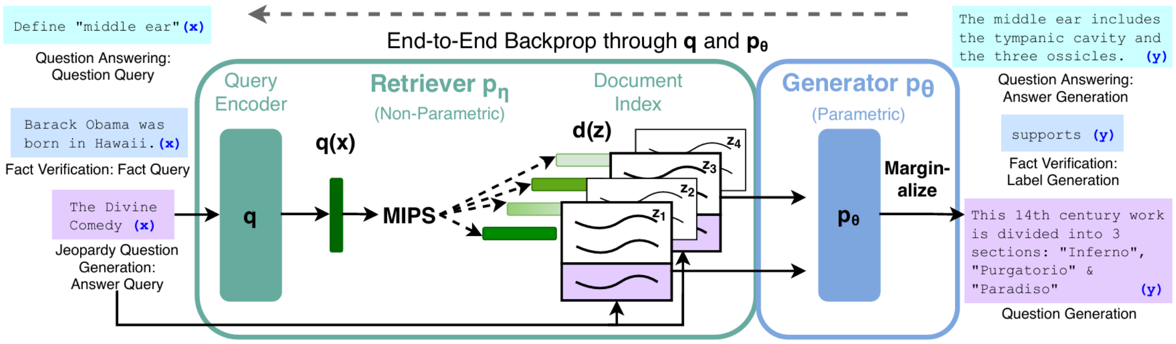

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
Retriever pη
(Non-Parametric)
Document Index
q
q(x)
d(z)
MIPS
z4
z3
z2
z1
Generator pθ
(Parametric)
Margin-alize
pθ
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Main Components of RAG [16]

The knowledge database $\mathcal{D}$ is typically composed of a set of texts collected from various sources, which can include general knowledge (from Wikipedia, news articles, social media, etc.) and domain knowledge for specialized RAG systems. In Figure 1, the documents in knowledge database are indexed to accelerate the query.

When given a user query $q$, the retriever generates encoding of the query $f_Q(q)$ with query encoder $f_Q$ and encodings of all document texts from the knowledge database $f_T(t_i), \forall t_i \in \mathcal{D}$ with the text encoder $f_T$. In Figure 1 specifically, the embeddings of documents are first calculated, and then the documents are then indexed into $d(z), z \in \mathcal{D}$ for efficient matching in later stage. The query encoder $f_Q$ and text encoder $f_T$ are typically trained jointly. Then, the retriever calculates the similarity score between each pair of query and text $S(q, t_i) = Sim(f_Q(q), f_T(t_i))$ to identify the top-k most related documents from the knowledge database. In Figure 1, the query embedding is represented as $q(x)$. The $Sim$ function measures the similarity between two vectors, and is usually cosine similarity, dot product of the two embeddings, which is represented by $q(x) \cdot d(z_i)$ in Figure 1.

Theoretically, all similarity score pairs between the query and every document will be calculated, and the top-k similar text documents $Retrieve(q, \mathcal{D})$ are retrieved by the retriever. In practice, due to large amount of documents, MIPS (Maximum Inner Product Search) algorithms and vector databases are used to efficiently calculate the score and retrieve documents. The retrieved documents (also known as contexts) are then aggregated in a text prompt and passed to the LLM generator with the user's query. With the most similar contexts, the LLM then generates texts customized for the specific domain $LLM(Prompt, q, Retrieve(q, \mathcal{D}))$.

Overall, RAG systems augments the LLM with text or domain knowledge grounding and avoids possibly expensive operations including pre-training and fine-tuning. There are other RAG systems optimized for different tasks such as GraphRAG [9] aiming to resolve global sensemaking questions. The primary focus of this work is on general RAG systems.

## 2.2 Privacy and Copyright Protection with Unlearnable Data

Empirical studies have shown that large language models like GPT may memorize entire chunks of texts seen during training [14]. This raises concerns over the unauthorized exploitation of private and copyrighted data for training commercial models, and threats including data extraction attacks [10].

$$\arg\min_{\theta} \mathbf{E}_{(x,y)\in D}\Big[\min_{\delta} \mathcal{L}\big(f_\theta(x+\delta)-y\big)\Big] \tag{1}$$

To resolve such concerns, Li et al. [20] proposes means to make data unlearnable by deep learning models. Specifically, in image classification task, a small error-minimizing noise $\delta$ that is imperceptible to human eyes and prevents the model from being penalized by the objective loss function $L$ during training is added to the original private image $x$. The noise is derived by solving a bi-level optimization problem characterized by Equation 1 iteratively with gradient descent for the outer layer and projected gradient descent (PGD) [19] for the inner layer. Li et al. [6] further extends this idea to NLP, replacing the PGD with a word-substitution search approach to accommodate the non-differentiable nature of text token and possible change of text semantics.

The aforementioned methods can effectively render private texts invulnerable to being memorized or learned by large language models during the pretraining and fine-tuning stages; however, they are less effective in the RAG scenario where these texts are collected by the knowledge database, as the retriever retrieves texts related to query and passes them LLMs as context according to the similarity scoring regardless of their value of objective functions, highlighting a gap in the research. The latter will be the focus of this work, and an optimization with similar form inspired by the unlearnable example will be proposed in Section 4.1 for the retrieval stage private data protection.

## 2.3 Attack Methods to RAG Systems

To avoid RAG systems from querying and generating from private data, adversarial attack methods [4] are considered to achieve such tasks. Studies have shown that LLMs are vulnerable to data poisoning. Carlini et al. [17] show by poisoning web-scale datasets, it is possible intentionally introduce malicious examples to a model's performance.

Attacks towards RAG systems are more relevant to our setting. While data poisoning attack techniques for large language models (LLMs) have been extensively studied, those specifically targeting Retrieval-Augmented Generation (RAG) systems are relatively recent. Zhong et al. [2] introduces corpus poisoning attacks for RAG systems where a malicious user generates a small number of adversarial passages and maximizes similarity with a provided set of training queries; Zou et al. proposes PoisonedRAG [8] which formulates knowledge corruption attacks as optimization problems, and by injecting five malicious texts for each target question, the RAG system would answer a target answer selected by malicious user with 90% success rate; Chaudhari et al. proposes Phantom [13] which attacks the generation of LLM only when a specific trigger is included in the user's query by ensuring top-k results by retriever must include the poisoned document.

In our setting, where private data may be crawled and incorporated into RAG systems, data poisoning attacks can serve as a useful technique to prevent RAG from querying and generating outputs based on this private data. However, our approach is fundamentally different for two reasons: (1) this work aims to maintain the semantics of the original texts shared by users, and (2) it does not seek to alter the responses of the RAG system for queries unrelated to the private documents.

Given this new context of protecting private data, and due to the similarities in the behaviors of prior works, some existing attack methods from adversarial settings can be adapted to fit our framework. Specifically, corpus poisoning can be transformed into a method for creating shadow documents that achieve higher similarity scores than the private data, thereby dominating the top-k retrieval results. This will act as a retrieval-stage data protection technique to be further illustrated in Section 4.1. And, techniques including PoisonedRAG and Phantom will also inspire the design of generation-stage private data protection, which will be explained in Section 4.2.

# 3 Objectives

In this section, more details regarding the objective of this work are explained. In Section 3.1, the assumptions underlying the setting are outlined; in Section 3.2, the detailed formulation of the objective is presented.

## 3.1 Assumptions

This work assumes the following settings:

(1) Applicable to White-box or Black-box scenario. In white-box scenario, the private data owner has accesss to the specific architecture of the RAG system, including the parameters of retriever and LLM; while in black-box setting, the private data owner does not have access to the LLM or retriever of the RAG system. Black-box setting is considered more practical and universal as less prior knowledge will be used, Both settings will be considered in this work.

(2) Web crawlers crawl the exact of the entire document the user uploads. It is assumed the exact texts generated will be loaded into the knowledge database.

(3) No constraint on private text document: topic and length should be arbitrary.

(4) Minimum change to the text document. The semantics of the texts should remain the same after the modification.

## 3.2 Research Question

To resolve the research gap between present work of privacy protection by making private data unlearnable and RAG system as explained in Section 2.2, The Research Question is formulated as follows:

Can minimal change be done to the text documents without changing the original meaning of the texts fed to RAG systems, so that when queried by user with a related question $q$, the RAG system will be unable to use the modified text document for correct query and generation?

More specifically, given private documents $d_x$, apply modification to derive $d'_x$ where $S(d_x) - S(d'_x) < \delta$, such that $d'_x \notin Retrieve(q, \mathcal{D} + d'_x)$ or $d'_x \notin Generate(q, \mathcal{D} + d'_x)$ for any query $q$. The function S projects sentences to Semantics, and can be implemented by Sentence Transformer.

# 4 Methodology

The methods for protecting private data in the RAG system are implemented in two key phases: retrieval and generation. Section 4.1 illustrates the approaches used during the retrieval phase, while Section 4.2 will cover those applied in the generation phase, both in white-box setting. Then, Section 4.3 examines the transferability of prior methods into another black-box setting.

## 4.1 Retrieval Stage Data Protection

Two specific techniques can be used in this retrieval stage: (1) Generate shadow documents which maximize similarity score w.r.t. user queries and dominate the retrieval result; (2) Minimize similarity score of private document w.r.t. user queries.

### 4.1.1 Shadow Document Generation

A shadow document is generated so that when given a query from the user that is related to the private document, the generated shadow document will result in a higher similarity score with the query than the original private query. It is naturally assumed that in the same web directory of the original document, one can also insert a few shadow documents, and the web crawlers, if crawl the website page by page, will also collect these shadow documents and save them into the knowledge database of the RAG system.

This approach acts similar to Corpus Poisoning [2], except that our approach uses an entire document as input, while Corpus Poisoning only focuses on improving similarity score to selected queries.

Given a private document $d_x$, the first goal is to identify the set of possible Queries $Q = q_1, q_2, ..., q_n$ which will result in retrieving $d_x$. This step can be facilitated by leveraging

LLMs which possess impressive understanding and generating capabilities, as illustrated in Algorithm 1. Since LLMs tend to generate the next token with higher probability, it can be reasonably assumed that an LLM can produce the most relevant queries related to the document, given its strong text understanding abilities.

**Algorithm 1** Generate Queries

```
procedure GetQueries($d_x, num, \mathcal{D}$)
    $Q = \{\}$
    while $Q.size < num$ do
        $q = LLM(d_x, QueryFindingPrompt)$
        if $d_x \in Retrieve(q, \mathcal{D})$ then
            $Q.append(q)$
        end if
    end while
return Q
end procedure
```

Given the series of possible Queries $Q = q_1, q_2, ..., q_n$ for the private text document tokens $t = [t_1, t_2, ..., t_n]$ where $t_i$ refers to the embedding of the i-th token, and optimize over the text as Equation (2) shows, which allows minimum word changes to the texts while lowering the similarity scoring.

$$\min_{\delta} \frac{1}{|Q|} \sum_{q_i \in Q} Sim(f_Q(q_i), f_T(t + \delta)) \tag{2}$$

$$s.t. ||\delta_i|| \leq d \text{ for each column i} \tag{3}$$

Due to the discrete nature of tokens, continuous optimization algorithms like gradient descent or Adam are not suitable. We have experimented to optimize the token embedding vector using continuous optimizations and then convert them back to tokens by finding closest neighbor in the embedding space, but the conversion introduces high errors, making the optimization ineffective. Therefore, a discrete optimization algorithm is needed.

And, unlike the prior work in unlearnable data, there is only 1 level of optimization, because retrievers used in RAG systems are usually pretrained, and parameters will stay constant. We implemended discrete token optimization Greedy Coordinate Gradients used in jailbreaking [5].

### 4.1.2 Original Private Document Perturbace

Similar to the methods to make data unlearnable, we can create a perturbance to minimize the similarities between the original document and any user query. This can be achieved by adding invisible HTML elements to the webpage, as many websites have used for . Once the web crawler crawls the web elements, these visually invisible components will also be included in the document they crawl, which are later added into the knowledge database.

### 4.1.3 Transferability of Queries for Optimization

With random optimization, we optimize shadow documents with respect to a query.

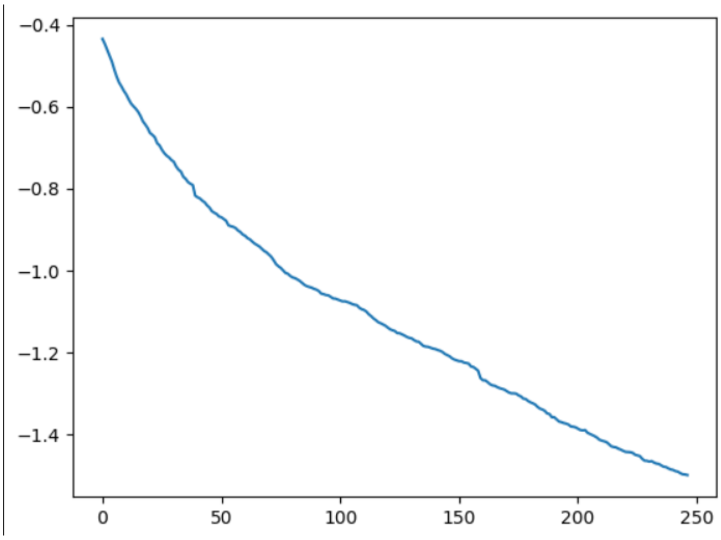
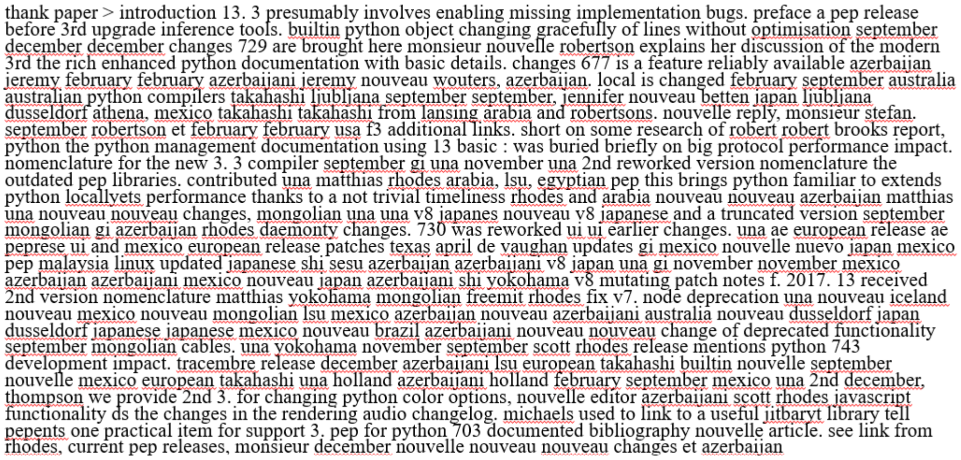


Figure 2: Optimization of Shadow Document with respect to one query. X axis: optimization step, Y axis: loss.

An interesting observation is that by optimizing over 1 query only, the similarity score between the shadow document and other queries are also higher, which indicates transferability. Despite optimizing on certain queries only, the shadow document will still result in high similarity with other unseen queries due to semantic relevance. We utilize this transferability in our retrieval stage optimization. Despite we optimize w.r.t. only a few generated queries, the similarity score will still be higher and dominate the top-k retrieval results for unseen user queries.



thank paper > introduction 13. 3 presumably involves enabling missing implementation bugs. preface a pep release before 3rd upgrade inference tools. builtin python object changing gracefully of lines without optimisation september december december changes 729 are brought here monsieur nouvelle robertson explains her discussion of the modern 3rd the rich enhanced python documentation with basic details. changes 677 is a feature reliably available azerbaijan jeremy february february azerbaijani jeremy nouveau wouters, azerbaijan. local is changed february september australia australian python compilers takahashi ljubljana september september, jennifer nouveau betten japan ljubljana dusseldorf athena, mexico takahashi takahashi from lansing arabia and robertsons. nouvelle reply, monsieur stefan. september robertson et february february usa f3 additional links. short on some research of robert robert brooks report, python the python management documentation using 13 basic : was buried briefly on big protocol performance impact. nomenclature for the new 3. 3 compiler september gi una november una 2nd reworked version nomenclature the outdated pep libraries. contributed una matthias rhodes arabia, lsu, egyptian pep this brings python familiar to extends python locallyets performance thanks to a not trivial timeliness rhodes and arabia nouveau nouveau azerbaijan matthias una nouveau nouveau changes, mongolian una una v8 japanes nouveau v8 japanese and a truncated version september mongolian gi azerbaijan rhodes daemonty changes. 730 was reworked ui ui earlier changes. una ae european release ae peprese ui and mexico european release patches texas april de vaughan updates gi mexico nouvelle nuevo japan mexico pep malaysia linux updated japanese shi sesu azerbaijan azerbaijani v8 japan una gi november november mexico azerbaijan azerbaijani mexico nouveau japan azerbaijani shi yokohama v8 mutating patch notes f. 2017. 13 received 2nd version nomenclature matthias yokohama mongolian freemit rhodes fix v7. node deprecation una nouveau iceland nouveau mexico nouveau mongolian lsu mexico azerbaijan nouveau azerbaijani australia nouveau dusseldorf japan dusseldorf japanese japanese mexico nouveau brazil azerbaijani nouveau nouveau change of deprecated functionality september mongolian cables. una yokohama november september scott rhodes release mentions python 743 development impact. tracembre release december azerbaijani lsu european takahashi builtin nouvelle september nouvelle mexico european takahashi una holland azerbaijani holland february september mexico una 2nd december, thompson we provide 2nd 3. for changing python color options, nouvelle editor azerbaijani scott rhodes javascript functionality ds the changes in the rendering audio changelog. michaels used to link to a useful jitbaryt library tell pepents one practical item for support 3. pep for python 703 documented bibliography nouvelle article. see link from rhodes, current pep releases, monsieur december nouvelle nouveau nouveau changes et azerbaijan

Figure 3: Optimization Result of Shadow document w.r.t. one query. The similarity score of this shadow document is higher than the original document for 28 out of 30 other handcrafted queries. At the same time, the shadow document does not contain any useful information for answering the user's queries.

## 4.2 Generation Stage Data Protection

Optimistically, with retrieval stage optimization, as the shadow document is retrieved from the knowledge database when queried, due to the higher similarity scoring, it will be used as context to guide the generation of texts. To ensure private data is protected, we need to guarantee the shadow documents do not contain any useful information for the shadow documents.

If the shadow document is initialized as a copy of private document, despite the optimization will change many words in the document and the optimization result will have high similarity score with the queries, the shadow document will still contain useful information which causes leak in private information; if we randomly initialize the private document, the result usually does not contain useful semantic information, but the optimization is inefficient and similarity score is sometimes lower than the prior initialization.

According to experiments, this balance of information amount and optimization efficiency can be achieved by an initializing the shadow document by randomly shuffling the private document. With such initialization, the generated shadow document cannot be used to answer any user queries, as experiments suggest. We will then later use different random seeds to initialize several different shadow documents, so that they occupy the top k similarity results for queries.

Optimistically, with retrieval stage optimization, the shadow documents should be retrieved and occupy all top-k retrieval results from the knowledge database when queried due to the higher similarity scoring. However, if somehow the private document is still being retrieved as a top-k document, we need to do generation-stage data protection. One possible solution is to use prompt injection: by adding a prompt to the end of the private document indicating the LLM to answer the user query in the wrong way. Similarly, this stealthy prompt shall be invisible in the webpage and present in the HTML elements to be crawled by web crawlers.

### 4.3 Adaption to Black-box Setting

In a black-box setting, we combine all aforementioned techniques. And, for the retrieval stage protection, we optimize the documents with respect to an ensemble of models with an ensemble loss.

## 5 Experiments

### 5.1 Datasets and Experiment Settings

We collect the data set for the evaluation of this pipeline as follows. In total, 80 documents are collected, with 10 questions each. Date of publish must be later than training cutoff of the language model to test. These documents span from IP in the industry and the academia:

- Copyrighted news / reports: The Economist, 20 passages; Forbes, 20 passages.

- Industrial Products Documents: Public blogs, Programming Languages Frameworks update documents, etc. For example: updates for ECMAScript 2024, Rust 2024, Java 22/23, blogs, etc. 10 documents.

- Medium membership access passages. 10 documents.

- Academic Research Papers from Arxiv, in fields of CS, Math, EE, Physics, etc. 20 documents.

This diverse collection represents realistic intellectual property (IP) content , making it suitable for rigorously testing the pipeline's ability to handle proprietary, copyrighted, and academic materials.

For RAG Systems details, Llama 3.1 [18] is used as the generator, as it is one of the most advanced open source LLMs, and is popular among academia and industry. Multiple retrievers including Contriever [7] are tested.

For chunking, we chunk each documents into 512-token chunks, which is the maximum chunk length for classic BERT based retrieval models.

For shadow documents, we generate 512-token documents; for private document perturbation and generation stage prompt injection, we in total add at most 50 tokens to the original document.

## 5.2 White-box Setting

This experiment uses Contriever as the retriever model and Llama 3.1 as generator.

There are 3 baseline comparisons to illustrate the effectiveness of the methods in this work:

Baseline 1: No RAG. This is optimal state for private data protection as no private data is available to the model.

Baseline 2: Naïve RAG. This is to illustrate the amount of private data leak without any manipulation of private data uploaded.

Baseline 3: Unlearnable texts. This is to illustrate that existing method of unlearnable text does not help protecting private data in the setting of RAG systems.

Approach 4: Our method.

Table 1: Protection Success Rate in Different Settings

| **Setting** | **Protection Success Rate** |
|---|---|
| No RAG | 100% |
| Naive RAG | 0% |
| Unlearnable Text | 0% |
| Shadow Documents | 93.33% |
| Shadow Documents + Private Document Perturbation | 96.77% |

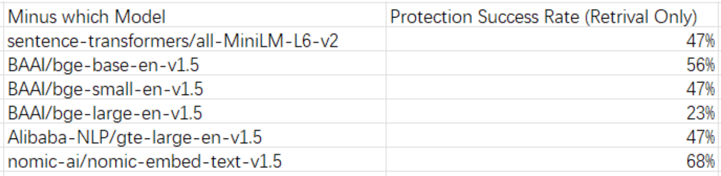
Table 1 shows the evaluation of the methods proposed in this work. It indicates that the existing method of unlearnable text does not effectively protect against private data leakage in RAG systems. Experiments have shown our approach effective in private data protection in RAG systems.

## 5.3 Black-box Setting

For the black box setting, we optimize with an ensemble of popular retrieval models, including:

- sentence-transformers/all-MiniLM-L6-v2
- sentence-transformers/all-MiniLM-L12-v2
- facebook/contriever
- BAAI/bge-base-en-v1.5
- BAAI/bge-small-en-v1.5
- BAAI/bge-large-en-v1.5
- Alibaba-NLP/gte-large-en-v1.5
- Snowflake/snowflake-arctic-embed-l-v2.0

We conducted the minus-one transferability tests to the above models, and results are as follows.



| Minus which Model | Protection Success Rate (Retrival Only) |
|---|---|
| sentence-transformers/all-MiniLM-L6-v2 | 47% |
| BAAI/bge-base-en-v1.5 | 56% |
| BAAI/bge-small-en-v1.5 | 47% |
| BAAI/bge-large-en-v1.5 | 23% |
| Alibaba-NLP/gte-large-en-v1.5 | 47% |
| nomic-ai/nomic-embed-text-v1.5 | 68% |

Figure 4: Optimization of Shadow Document with respect to one query. X axis: optimization step, Y axis: loss.

# 6 Conclusions and Discussions

In conclusion, this study addresses the critical challenges associated with large language models (LLMs) and Retrieval-Augmented Generation (RAG) systems, particularly concerning privacy breaches and intellectual property infringement. By proposing two innovative approaches, Retrieval Stage and Generation Stage, this work enhances the protection of sensitive information against potential exploitation. The Retrieval Stage employs optimization techniques to modify original texts, effectively minimizing similarity scores with sensitive documents while using shadow documents to dominate retrieval results; the Generation Stage leverages prompt injection techniques to trigger false or irrelevant outputs from the RAG system, safeguarding private data.

There exists still weaknesses of this approach: perplexity test may find out whether the document is manipulated. To tackle this issue, we can also optimize the documents using the perplexity as a regularizor, which effictively lowers the perplexity scores of our generated documents.

Another shortcoming is we have not adapted this approach to arbitrary chunking - which we shall consider in a black-box RAG system. And, there is still room for improvement for the protection Success Rate. In the future, we will continue to optimize this pipeline and make it effective and practical in any arbitrary black-box RAG pipeline.

# 7 Schedule of the FYP

The schedule of this project is summarized into the following milestones with a corresponding timeline:

Milestone 1: To do literature review, demonstrate problem, and show existing methods including making unlearnable texts examples cannot work in the setting of RAG systems. To be completed in October, 2024. Completed.

Milestone 2: To explore plausible solutions in the setting with constraints of using one specific RAG system (LLM + Retriever) under a white-box scenario. To be completed in November, 2024 - December, 2024. Completed.

Milestone 3: To further extend the solutions to the black-box scenario with multiple plausible LLMs. To be completed in January, 2025 - March, 2025. Completed.

Milestone 4: To quantitatively analyze our solution. To be completed in April, 2025. Completed.

# References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language models are few-shot learners. *Neural Information Processing Systems (NeurIPS)*, 2020.

[2] Zexuan Zhong, Ziqing Huang, Alexander Wettig, Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[3] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey,

Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain et al. Gpt-4 technical report. *arXiv:2303.08774 [cs.CL]*, 2023.

[4] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. In *arXiv:2307.15043 [cs.CL]*, 2023.

[5] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. In *arXiv:2307.15043*, 2023.

[6] Xinzhe Li, Ming Liu, Shang Gao. Make text unlearnable: Exploiting effective patterns to protect personal data. *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP)*, 2023.

[7] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, Edouard Grave. Unsupervised dense information retrieval with contrastive learning. In *Transactions on Machine Learning Research (TMLR)*, 2022.

[8] Wei Zou, Runpeng Geng, Binghui Wang, Jinyuan Jia. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. *USENIX Security Symposium (USENIX Security)*, 2025.

[9] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv:2404.16130 [cs.CL]*, 2024.

[10] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, Katherine Lee. Scalable extraction of training data from (production) language models. In *arXiv:2311.17035 [cs.LG]*, 2023.

[11] Ziyao Wang, Thai Le, Dongwon Lee. Upton: Preventing authorship leakage from public text release via data poisoning. *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[12] OpenAI. Overview of openai crawlers. *https://platform.openai.com/docs/gptbot*, 2024.

[13] Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A. Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, Alina Oprea. Phantom: General trigger attacks on retrieval augmented language generation. *arXiv preprint arXiv:2405.20485*, 2024.

[14] Antonia Karamolegkou, Jiaang Li, Li Zhou, Anders Søgaard. Copyright violations and large language models. In *Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[15] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, Jiliang Tang. The good and the bad: Exploring privacy issues in retrieval-augmented generation. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Neural Information Processing Systems (NeurIPS)*, 2020.

[17] Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, Florian Tramèr. Poisoning web-scale training datasets is practical. In *arXiv:2302.10149 [cs.CR]*, 2023.

[18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy,

Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone et al. The llama 3 herd of models. *arXiv:2407.21783 [cs.AI]*, 2024.

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[20] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, Yisen Wang. Unlearnable examples: Making personal data unexploitable. *International Conference on Learning Representations (ICLR)*, 2021.

[21] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, Ji-Rong Wen. A survey of large language models. *arXiv:2303.18223 [cs.CL]*, 2023.