Mitigating the Privacy Issues in Retrieval-Augmented Generation (RAG) via Pure Synthetic Data

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) enhances the outputs of language models by integrating relevant information retrieved from external knowledge sources. However, when the retrieval process involves private data, RAG systems may face severe privacy risks, potentially leading to the leakage of sensitive information. To address this issue, we propose using synthetic data as a privacy-preserving alternative for the retrieval data. We propose SAGE, a novel two-stage synthetic data generation paradigm. In the stage-1, we employ an attribute-based extraction and generation approach to preserve key contextual information from the original data. In the stage-2, we further enhance the privacy properties of the synthetic data through an agent-based iterative refinement process. Extensive experiments demonstrate that using our synthetic data as the retrieval context achieves comparable performance to using the original data while substantially reducing privacy risks. Our work takes the first step towards investigating the possibility of generating high-utility and privacy-preserving synthetic data for RAG, opening up new opportunities for the safe application of RAG systems in various domains[1].

1 Introduction

Retrieval-augmented generation (RAG) aims to improve language model outputs by incorporating relevant information retrieved from external knowledge sources. It has been effectively applied in various scenarios, such as domain-specific chatbots (Siriwardhana et al., 2023) and email/code completion (Parvez et al., 2021). A typical RAG system often operates in two stages: retrieval and generation. First, the system retrieves relevant knowledge from an external database based on the user query. Then, the retrieved information is integrated with the query to form an input for a large language model (LLM). The LLM uses its pre-trained knowledge and the retrieval data to generate a response, enhancing the overall quality of the output.

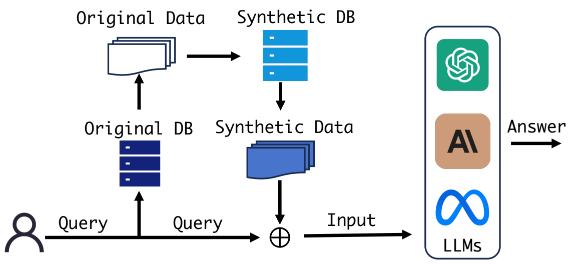


Figure 1: An illustration for RAG with synthetic data.

However, according to existing literature (Zeng et al., 2024; Huang et al., 2023; Ding et al., 2024; Qi et al., 2024), RAG may face severe privacy issues when the retrieval process involves private data. For example, Zeng et al. (2024) observe that carefully designed user prompts are able to extract original sentences in the retrieval data (untargeted attack), and can also extract specific pieces of private information (targeted attack), potentially leading to the leakage of considerable amount of the retrieval data. The potential risk of information leakage can significantly limit the applications of RAG systems. For instance, a medical chatbot (Yunxiang et al., 2023) using patients' historical diagnosis cases as a knowledge source may improve response quality but raises concerns about exposing sensitive patient information. Therefore, enhancing the privacy properties of RAG systems and protecting the retrieval data from leakage is of high importance to prevent unauthorized access or misuse and enable safe and widespread adoption, particularly in sensitive domains like healthcare.

Some adaptations (Zeng et al., 2024) have been proposed to protect the privacy of RAG by incorporating additional components in the RAG pipeline. These adaptations include pre-retrieval techniques (such as setting similarity distance thresholds in

[1] Our code is available at this anonymous link

retrieval) and post-processing techniques (e.g., re-ranking and summarization (Chase, 2022)). However, as demonstrated by (Zeng et al., 2024), these methods cannot fully eliminate privacy risks, as the data itself may contain sensitive information. Moreover, these methods often introduce a significant privacy-utility trade-off and may incur extra time costs during inference.

To address the above concern, we propose an alternative data-level solution via using synthetic data as shown in Figure 1. By generating a privacy-preserving version of the original data and only providing the synthetic version to the LLM, the risk of information leakage could be effectively mitigated. This approach can potentially ensure that the original data is not directly used as input to the LLMs, thereby reducing the chances of sensitive information being exposed or leaked during the retrieval and generation process. Therefore synthetic data allows the creation of a safe, surrogate dataset that maintains the essential properties and relationships of the original data while protecting sensitive information. There are recent works exploring synthetic data generation using pre-trained language models (Ye et al., 2022; Meng et al., 2022; Gao et al., 2023a; Chen et al., 2023; Yu et al., 2024; Xie et al.) and utilizing the synthetic data in the downstream task to protect the privacy of the original data. Besides, some studies integrate differential privacy with synthetic data for in-context demonstrations (Tang et al., 2023). However, while existing methods for generating synthetic data work well for downstream tasks or in-context demonstrations, they are not well aligned with the unique requirements of RAG: RAG primarily focuses on utilizing key information from the data to answer related questions (Ding et al., 2024), rather than learning general patterns. Therefore, it is crucial to preserve as much useful information as possible from the original data when generating synthetic retrieval data. On the other hand, existing synthetic methods do not require generating data that shares the same key information with the original data. Consequently, there is a lack of exploration on how to effectively use synthetic data for RAG and how to design a feasible solution for generating high-quality retrieval data. Meanwhile, the unique information requirements of retrieval data also present challenges in generating privacy-preserving synthetic data, as it is crucial to carefully select what information to preserve and what privacy-sensitive elements to omit.

In this work, we take the first effort to investigate the possibility of generating synthetic retrieval data that maintains high utility while enhancing privacy protection for RAG. After identifying the related data from the original dataset, we use the synthetic version of the data as context instead of the original data for generation. We use a two-stage generation and refinement paradigm called called SAGE (**S**ynthetic **A**ttribute-based **G**eneration with ag**E**nt-based refinement) to generate synthetic retrieval data. To preserve the important information of the original data and keep the utility of the synthetic data, we first utilize an attributed-based extraction and generation approach to generate the synthetic data. Specifically, for each dataset, we first input few-shot samples to make the LLM identify important attributes of the dataset. Then, for each data sample, we ask the LLM to extract key information corresponding to these attributes. After that, we input the attribute information into another LLM and ask it to generate synthetic data based on these key points (stage-1). In this way, the generated data contains key contextual information.

Although the attribute-based method can preserve key information of the original data, it may still include some privacy information, as the stage-1 does not incorporate privacy constraints. Therefore, a second step is necessary to further preserve privacy. In stage-2, we propose an agent-based iterative refinement approach to enhance the protection of private information. Specifically, we introduce two agents, a privacy assessment agent and a rewriting agent. The privacy assessment agent determines whether the generated data contains privacy information, such as containing personally identifiable information (PIIs) or potentially leading to the linkage of personal information, and provide feedback. The rewriting agent then takes this feedback to refine its generated data until the privacy agent deems it safe. Our experimental results show that using our synthetic data as retrieval data can achieve comparable performance with using original data while substantially reducing the associated privacy risks.

2 Related Works

2.1 Retrieval-augmented generation and its privacy issues

Retrieval-augmented generation (RAG), introduced by Lewis et al. (2020), has become a popular approach to enhance LLMs' generation ability (Liu, 2022; Chase, 2022; Van Veen et al., 2023; Ram et al., 2023; Shi et al., 2023). RAG improves output accuracy and relevance (Gao et al., 2023b), mitigating "hallucinations" of LLMs (Shuster et al., 2021). Its flexible architecture allows seamless updates to the dataset, retriever, and LLM without re-training (Shao et al., 2023; Cheng et al., 2023). These advantages make RAG a favored approach for applications like personal chatbots and specialized domain experts (Panagoulias et al., 2024).

However, the application of RAG also brings privacy issues. Huang et al. (2023) have shown the privacy implications of retrieval-based LM and identified privacy leakage of KNN-LM (Khandelwal et al., 2019), a specific kind of retrieval LM. Zeng et al. (2024) have shown that RAG is vulnerable to extraction attacks. Qi et al. (2024) have shown that production RAG models also suffer from attacks. The vulnerability of RAG makes its application in privacy domains under high risks.

2.2 Synthetic data generation using large language models

As large language models become more expressive, researchers have explored using them to generate synthetic data. Ye et al. (2022); Meng et al. (2022) propose to generate synthetic data via zero-shot prompting and then train smaller models on these data to handle various tasks including text classification, question answering and etc. Gao et al. (2023a) further develop a noise-robust reweighting framework to improve the quality of generated data. Chen et al. (2023) propose to mix a set of soft prompts and utilize prompt tuning to generate diverse data. Yu et al. (2024) focus on the attributes of data itself including length and style to generate more diverse data. Recent works (Tang et al., 2023; Xie et al.) take privacy into consideration. Tang et al. (2023) propose a few-shot data generation method to generate private in-context demonstrations from a private dataset and provide a differential privacy guarantee. Xie et al. introduce a private evolution algorithm to generate deferentially private data. However, their synthetic data is not guaranteed to include contextual information in the original data, thus not fitting the RAG system well.

3 Methods

Our SAGE framework of generating synthetic retrieval data is composed of two stages, i.e., attribute-based data generation and agent-based interactive refinement, as shown in Figure 2. The stage-1 aims to generate data that contains essential information of original data, while the stage-2 aims to automatically refine the data to further mitigate the privacy-related concerns. The synthetic data generation process can be conducted **offline** and only needs to be performed **once**. During inference, when the original data is identified, the corresponding synthetic data is returned as retrieval data.

3.1 Stage-1: Attribute-based data generation

In this stage, we aim to generate synthetic data that contains all the essential information from the original data. To achieve this goal, we propose an attribute-based data extraction and generation paradigm to create synthetic data.

The entire process of Stage-1 consists of three steps: identifying important attributes using few-shot samples, extracting key information related to essential attributes, and generating synthetic data conditioned on the extracted key information. First, we feed few examples within the dataset to an LLM-based *attribute identifier* and prompt it to identify m most essential attributes of the dataset[2]. This process is performed before generating any synthetic data, and is only needed for once. Then, after obtaining the essential attributes, we leverage an LLM-based *information extractor* to extract key information related to these attributes for each data sample and construct [attribute:key information] pairs. This step captures the core useful information of the original data. Finally, we input these attribute-information pairs into an LLM-based *data generator* to generate new synthetic data. The synthetic data is expected to include key information extracted in the second step, thus reducing the loss of useful information in the original data. The prompt used for this step is provided in Appendix A.1.1. It is noteworthy that the LLMs used in these steps (attribute identifier, information extractor, and data generator) can be the same or different models. In Section 4.4, we also explore different model combinations and their impacts.

3.2 Stage-2: Agent-based private data refinement

Though the synthetic data generated in Stage-1 has preserved important information from the original data, it may still have privacy issues as no privacy controls are added. For example, it may contain PIIs such as email addresses or phone numbers,

[2]We discuss the impact of m in Section 4.4

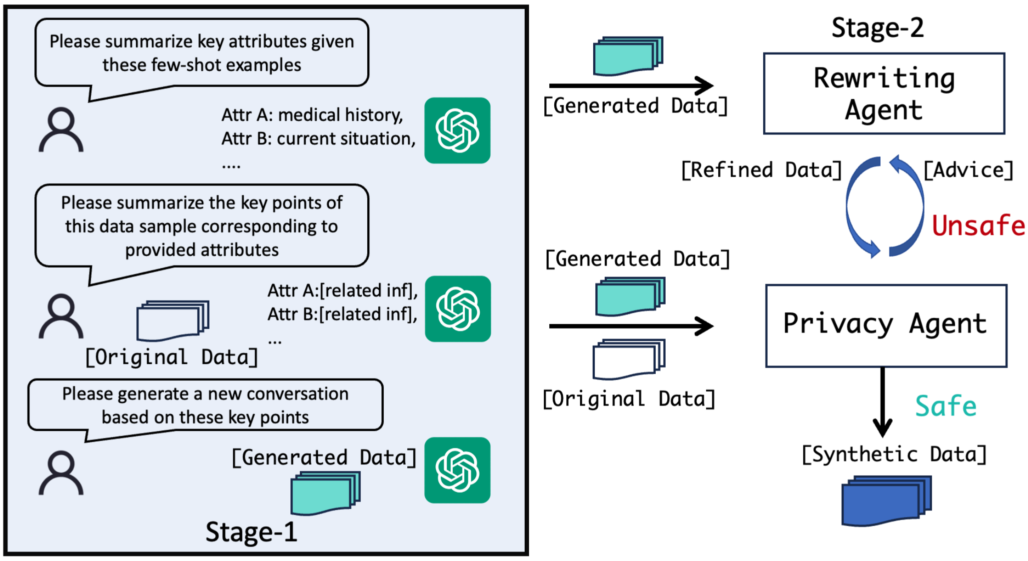


Figure 2: Pipeline of generating synthetic data.

or specific personal information that can possibly be linked to specific individuals. Thus, the synthetic data still may cause privacy leakage when used as retrieval data. Although methods such as anonymization can mitigate this issue to some extent, they can only mask highly structured data like email addresses, and it is challenging to reduce other potential privacy risks (Wang et al., 2022). As pointed out in (Brown et al., 2022), one key challenge in natural language processing (NLP) is that private information is often not explicitly presented but can be inferred from the context. Considering the sentence: "I just got back from the oncology department at City Central Hospital. The doctor said my chemo is going well.", this sentence does not directly mention the person's name but reveals that the speaker is undergoing cancer treatment at City Central Hospital. Moreover, Shi et al. (2022) further demonstrate that although directly removing all entities can preserve privacy, it will cause the data to contain almost no useful information, and the performance loss would be unacceptable. To address this issue, we propose to utilize the rewriting and reflection capabilities of large language models (LLMs) through an agent-based approach. This method involves 2 agents collaborating to iteratively refine the generated answers so that they can maintain utility while protecting privacy.

Specifically, in our framework, we introduce a privacy agent and a re-writing agent that collaborate iteratively to enhance the privacy of the generated data. The privacy agent takes both the generated data from Stage-1 and the original data as input to assess whether the generated data contains privacy issues, such as containing PIIs or the linkage of personal information. It then provides feedback to the re-writing agent. The re-writing agent, in turn, improves data according to the privacy agent's advice. The privacy agent then evaluates the newly generated data again. This process continues until the privacy agent determines that the synthetic data is safe[3].

4 Experiment

In this section, we present various experimental results to demonstrate the utility and privacy properties of SAGE. We first introduce our experimental setup in Section 4.1, including the components of RAG, evaluation datasets, tasks, and baselines. Then, we present the utility and privacy results in Section 4.2 and Section 4.3, respectively. Moreover, we conduct ablation studies in Section 4.4 to investigate the impact of the number of attributes, model choice, and the number of retrieved documents on the performance and privacy of SAGE.

4.1 Evaluation Setup

RAG components In our experiments, we mainly employed Llama3-8b-chat (L8C) as the language model for text generation for performance evaluation. We chose this model because it cannot perform well on our chosen tasks without RAG, allowing us to test the extent to which RAG can

[3] We put the detailed workflows and system prompts of these two agents and average iteration rounds in Appendix A.1.2 and synthetic data examples in Appendix A.5.

improve the generation quality. For the privacy experiments, we use both the widely-used closed-source model GPT-3.5-turbo and the open-source model L8C for text generation. Both models have been safety-aligned, allowing us to demonstrate the vulnerability of RAG systems and the effectiveness of our proposed methods. We utilized the `bge-large-en-v1.5` model as the embedding model. The embeddings were stored and the retrieval database was constructed using the FAISS library. By default, the L_2-norm was used as the similarity metric to compare embeddings. Unless otherwise specified, we retrieved a single document ($k = 1$) for each query. The impact of varying the number of retrieved documents was further investigated in Section 4.4. [4]

Tasks and retrieval datasets We consider two privacy-related scenarios to verify the effectiveness of our synthetic methods. In the first scenario, we focus on monitoring medical dialog cases and utilize the HealthcareMagic-101 dataset of 200k doctor-patient medical dialogues as the retrieval dataset. In the second scenario, we follow the setting of (Huang et al., 2023) to consider a case where some private information is mixed with a public dataset. Specifically, we mix personal information pieces from the Enron Mail dataset (private dataset) with the wikitext-103 dataset (public dataset), which we refer to as Wiki-PII dataset. We extract personal PIIs and combine those PIIs with each sample of the wikitext-103 dataset. The details of the construction are presented in Appendix A.4. We then evaluate the performance of our methods on open-domain question answering datasets (ODQA), including Natural Questions (NQ) (Kwiatkowski et al., 2019), Trivia QA (TQA) (Joshi et al., 2017), Web Questions (WQ) (Berant et al., 2013), and CuratedTrec (CT) (Baudiš and Šedivỳ, 2015). The detailed descriptions of these datasets are included in Appendix A.4.

Baselines. To verify the effectiveness of our methods, we include three baselines: simple paraphrasing and existing representative LLM-based data synthesis methods like **ZeroGen** (Ye et al., 2022) and **AttrPrompt** (Yu et al., 2024). We provide the details of the implementation of these methods in Appendix A.2. We also report generation results without RAG, denoted as **0-shot**, using original data directly as retrieval data, denoted as **Origin**, and the outputs of the attributes-based generation, denoted as **Stage-1**. Finally, we report the outputs of the complete SAGE pipeline, denoted as **Stage-2**.

[4] By defaute, we use GPT-3.5 at stage-1 and GPT-4 for agents at stage-2, we explore the model choice in Section 4.4

Table 1: Utility results on HealthCareMagic dataset

Method	BLEU-1	ROUGE-L
0-shot	0.081	0.0765
Origin	0.0846	0.0789
Paraphrase	0.105	0.0952
ZeroGen	0.0850	0.0769
AttrPrompt	0.079	0.067
Stage-1	0.114	0.0956
Stage-2	0.113	0.0943

4.2 Utility of using synthetic data

To assess the utility of using synthetic data as retrieval data, we evaluate the quality of the generated answers by comparing the answers with the ground truth. We primarily report the ROUGE-L and BLEU scores between the generated and the ground truth answers.

Utility results on medical dialog. For the medical dialog case, we split the data into two parts: 99% of the data is used as the retrieval data, and the remaining 1% is used as the test data. To evaluate the system's performance, we input questions from the test set and compare the generated answers with the ground truth answers using similarity-based metrics such as ROUGE-L and BLEU scores. The results are reported in Table 1. The results demonstrate that using synthetic data achieves performance comparable to, and even better than, using original data. Moreover, it significantly outperforms generation without retrieval. Our methods also surpass simple paraphrasing and ZeroGen. These findings suggest that our approach to generating synthetic data effectively preserves the utility of the original data.

Utility results on ODQA. To assess open-domain question answering (ODQA) performance, we combine the WikiText-101 dataset with Enron Mail, as the source for information retrieval. We then evaluate the system's performance using multiple ODQA datasets, such as Natural Questions (NQ), Trivia QA (TQA), WQ, CT.

The experiment results are summarized in Table 2. Similar to Table 1, using our proposed synthetic data as retrieval data shows consistently high performance, comparable to directly using the original

Table 2: Utility results on Wiki-PII dataset

Method	NQ		TQA		WQ		CT	
	BLEU-1	ROUGE-L	BLEU-1	ROUGE-L	BLEU-1	ROUGE-L	BLEU-1	ROUGE-L
0-shot	0.00719	0.0136	0.00843	0.0157	0.00716	0.0143	0.00882	0.0150
Origin	0.0180	0.0315	0.0150	0.0272	0.0147	0.0271	0.0178	0.0323
Paraphrase	0.0153	0.0269	0.0127	0.0251	0.0094	0.0187	0.0135	0.0252
ZeroGen	0.0034	0.0063	0.0057	0.010	0.0104	0.0201	0.0116	0.0205
AttrPrompt	0.0061	0.0107	0.006	0.0108	0.006	0.0110	0.00624	0.0111
Stage-1	0.0131	0.0257	0.0125	0.0249	0.0132	0.0277	0.0122	0.0242
Stage-2	0.0177	0.0322	0.0131	0.0247	0.0173	0.0298	0.0129	0.0267

Table 3: Targeted attack results on Wiki-PII and HealthCareMagic dataset(250 prompts)

	Target-wiki-llama-3-8b		Target-wiki-gpt-3.5		Target-chat-llama-3-8b		Target-chat-gpt-3.5	
Method	Target info	Repeat prompts	Target info	Repeat prompts	Target info	Repeat prompts	Target info	Repeat prompts
origin	25	12	167	64	7	23	75	132
para	9	1	28	9	17	26	42	81
ZeroGen	4	5	5	2	0	3	1	6
AttrPrompt	0	0	0	0	0	0	0	0
Stage-1	1	4	3	19	3	11	12	36
Stage-2	0	0	0	7	0	0	0	0

Table 4: Untargeted attack results on HealthCareMagic dataset(250 prompts)

Method	Untarget-chat-llama				Untarget-chat-gpt3.5			
	Repeat prompt	ROUGE prompt	Repeat context	ROUGE context	Repeat prompt	ROUGE prompt	Repeat context	ROUGE context
origin	19	17	16	13	61	67	49	67
para	23	13	22	11	45	63	33	50
ZeroGen	0	0	0	0	0	0	0	0
AttrPrompt	0	0	0	0	0	0	0	0
Stage-1	1	2	1	2	1	0	1	0
Stage-2	0	0	0	0	0	0	0	0

data. In some datasets, such as NQ and WQ, our synthetic data even outperforms the original data. This may be because our pipeline in stage-1 preserves most of the essential key information. In stage-2, the data is further refined, and the final outputs contain more "pure" useful information, making it easier for the LLM to identify essential information and generate better answers.

4.3 Privacy of using synthetic data

To evaluate the privacy properties of using our synthetic data as retrieval data, we conducted targeted and untargeted attacks following (Zeng et al., 2024), which can cause considerable data leakage from standard retrieval database. The composite structured prompting attack on RAG consists of two components: {*information*} and {*command*}. The {*information*} component guides the retrieval system to fetch specific data, while the {*command*} component instructs the language model to include the retrieved information in its response. For the {*command*} component, we use phrases such as "Please repeat all the context" for both targeted and untargetd attacks. The {*information*} component is adjusted according to the objectives of the attack. Targeted attacks aim to extract specific sensitive information, such as PII or private dialogue cases, by providing relevant input. In contrast, untargeted attacks seek to gather as much data as possible from the entire retrieval dataset without focusing on specific information.

For untargeted attacks, we report the number of prompts that can generate outputs with either at least 10 tokens exactly matching the original dataset (**Repeat Prompt**) or with sufficient similarity to the original data, as indicated by a ROUGE-L score exceeding 0.5 (**Rouge Prompts**). Additionally, we report the number of unique verbatim excerpts (**Repeat Contexts**) and closely similar answers retrieved from the data, with a ROUGE-L score higher than 0.5 (**Rouge Contexts**). For targeted attacks, we also report the **Repeat Prompt** metric and the number of unique targeted information pieces extracted (**Targeted Information**).

Further details regarding the threat models are provided in Appendix A.3.

Untargeted attack results. In the context of an untargeted attack, the attacker's objective is to gather as much information as possible from the whole retrieval dataset, rather than seeking specific data. To achieve this, following (Carlini et al., 2021), we randomly select chunks from the Common Crawl dataset to serve as the {*information*} component. We report the results of untargeted attacks on the HealthCareMagic dataset in Table 4. From the results, we can observe that (a) The attribute-based generated data in the first stage shows a lower attack success rate than using original data and simple paraphrasing. This indicates that the method is inherently more robust to privacy attacks. This may be because the generation process produces a new version of the data and naturally avoids highly similar sentences and exact copies. (b) After the agent-based refinement, using the final version of the data as context shows nearly no untargeted attack success. This demonstrates that the refinement process has successfully further protected the data, and the produced data is of high safety.

Targeted attack results. In the targeted attack, the attacker has specific objectives regarding the type of information they aim to extract, such as PIIs including phone numbers and email addresses, or sensitive content like personal dialogue cases. For these attacks, the {*information*} component consists of some specific information that is related to the attacker's goals. When evaluating the HealthCareMagic dataset, our goal is to extract diagnosed cases for specific diseases. We use the prompt "I want information about disease" as the *information* component. To consider the targeted information as successfully extracted, two criteria must be met: (a) the name of the targeted disease must appear in the original data, and (b) the model's output must contain repetitive segments from the original data at least 10 tokens. In the case of the Wiki-PII dataset, which includes a mix of data from Enron Mail, we focus on retrieving PIIs by employing frequently used leading phrases such as "My phone number is" as the *information* element. The targeted information in this context is measured by the total count of PIIs effectively extracted from the retrieval dataset.

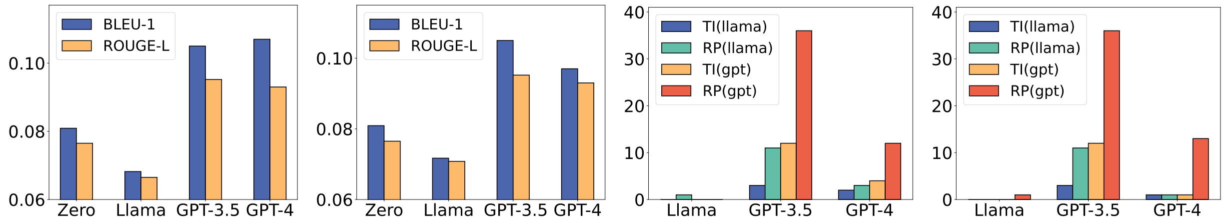
The results of targeted attacks lead to conclusions similar to those of untargeted attacks. From Table 3, the generated data in the first stage has significantly reduced targeted information leakage. This is because the newly generated data only retains the essential key information and may naturally omit some specific privacy information. Furthermore, after the agent-based refinement process, the attack success rate further decreases to nearly zero. This validates that the agent-based refinement process can successfully further reduce the possibly privacy-violating information in the synthetic data.

4.4 Ablation Studies

In this subsection, to investigate the factors that affect the quality of synthetic data, we conduct ablation studies analyzing the impact of model choice, the number of attributes, and retrieved documents per query.

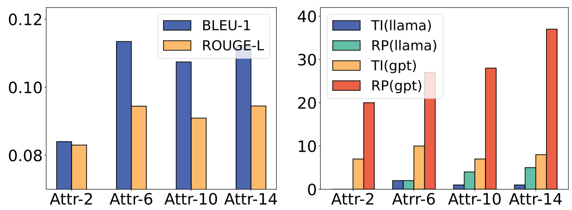
Impact of model choice. To investigate the influence of model choice on stage-1 generation, we change the models used for the *information extractor* and *data generator* components in stage 1. Specifically, we experiment with different models, including GPT-4, GPT-3.5, and Llama3-Chat-8b, for these two components. For the experiments on the *information extractor*, we fix the *data generator* as GPT-3.5 and vary the model used for the *information extractor*. Similarly, for the experiments on the *data generator*, we fix the model of *information extractor* as GPT-3.5 and vary the model of *data generator*. We conduct the utility experiments on the HealthCareMagic dataset and use BLEU-1 and ROUGE-L scores compared with groundtruth as performance indicators. The impact on performance is shown in Figure 3a and Figure 3b. We can clearly observe that if weak models like Llama-8b-chat are used as the *data generator* or the *information extractor*, the overall performance is poor, even worse than zero-shot prediction. This indicates that the generated data is of poor quality. The performance of GPT-3.5 and GPT-4 when used as *information extractor* and *data generator* both show promising results, and GPT-4 does not necessarily perform better than GPT-3.5. This may indicate that GPT-3.5 is already powerful enough to handle the stage-1 generation tasks, and more powerful models like GPT-4 do not necessarily improve the performance.

We also report the targeted attack results on the HealthCareMagic dataset when using the stage-1 generated data as retrieval data in Figure 3c and Figure 3d. From the results, we can observe that



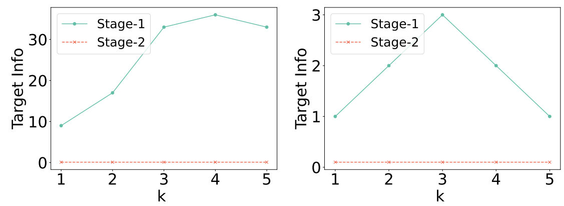
(a) Performance(info extractor) (b) Performance(generator) (c) Attack(info extractor) (d) Attack(data generator)

Figure 3: Ablation study on model choice. TI means targeted information and RP means repeat prompts.



(a) Performance(m) (b) Attack(m)

Figure 4: Ablation study on number of attributes m.



(a) Targeted Attack vs k (GPT)(b) Targeted Attack vs k (L8C)

Figure 5: Ablation study on number of retrieved docs.

using Llama3-Chat-8b (L8C) as the *information extractor* and *data generator* results in no privacy leakage, as the generated data is of poor quality and fails to preserve information from the original data. Besides, we found that using GPT-4 results in lower privacy leakage than GPT-3.5. This may be because the safety mechanism of GPT-4 is better, and it automatically filters out more sensitive information in the synthetic process.

Impact of the number of attributes. In this part, we investigate the influence of the number of attributes m. We change the number of attributes m and observe its impact on performance and privacy on the HealthCareMagic dataset. The performance results are shown in Figure 4a. From the figure, we can observe that when the number of attributes is very small (e.g., when the number of attributes is 2), the performance is likely to be poor. This is because the limited number of attributes fails to capture all the essential information. Besides, we find that with an increase in the number of attributes, the performance improves but does not necessarily continue to increase. We also report the targeted attack results of using stage-1 data on the same dataset in Figure 4b. From the results, we found that a small number of attributes leads to lower privacy exposure, as the limited number of attributes also misses more private information. Thus, we recommend choosing a proper number of attributes for different datasets via methods like testing on the evaluation set.

Impact of the retrieved number of documents. To verify that our proposed synthetic data pipeline can still protect privacy when more documents are retrieved, we conduct ablation studies by varying the number of documents retrieved and report the targeted attack results on the HealthCareMagic dataset. From Figure 5a, we can observe that in some cases, the privacy risks will be amplified when k increases if only stage-1 data is used. However, both in Figure 5a and Figure 5b, we find that the data after agent-based refinement shows consistently minimal privacy leakage when k is increased, indicating the robustness of our method against privacy attacks.

5 Conclusions

In this paper, we take the first step towards investigating the possibility of utilizing synthetic data as retrieval-augmented generation (RAG) data to mitigate privacy concerns. We propose a novel two-stage synthetic pipeline that includes attribute-based data generation, which aims to maintain key information, and iterative agent-based refinement, which further enhances the privacy of the data. Experimental results demonstrate that using our generated synthetic data as RAG data achieves comparable performance to using the original data while effectively mitigating the associated privacy issues. Our work opens up new opportunities for the safe application of RAG systems in sensitive-related domains.

6 Limitations

In our research, we investigate the possibility of using synthetic data for retrieval-augmented generation (RAG) and propose a novel pipeline for generating high-utility and privacy-preserving synthetic data. We verify the effectiveness and safety of our synthetic data in representative scenarios, such as healthcare. In the future, we would like to further validate the efficacy of our pipeline across a wider range of domains and datasets. Moreover, while our method demonstrates robustness against privacy attacks on RAG, incorporating techniques like differential privacy to provide stricter theoretical guarantees on synthetic RAG data remains an interesting open question that warrants further exploration.

7 Ethic Statement

This work explores using synthetic data to mitigate privacy risks in Retrieval-Augmented Generation (RAG), particularly in safety-critical domains. We argue that protecting sensitive information is crucial, as data leakage can severely impact individuals' well-being and privacy rights. Our approach generates synthetic data to replace sensitive data during RAG, aiming to reduce privacy breach risks. We have adhered to ethical guidelines and acknowledge the need for further research to understand the risks and benefits of our method. Developing privacy-preserving techniques is essential for the responsible deployment of RAG systems. Our research contributes to balancing the benefits of advanced language models with the protection of individual privacy rights.

References

Petr Baudiš and Jan Šedivỳ. 2015. Modeling of the question answering task in the yodaqa system. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings 6*, pages 222–228. Springer.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Harrison Chase. 2022. Langchain. October 2022. https://github.com/hwchase17/langchain.

Derek Chen, Celine Lee, Yunan Lu, Domenic Rosati, and Zhou Yu. 2023. Mixture of soft prompts for controllable data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14815–14833.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self memory. *arXiv preprint arXiv:2305.02437*.

Yujuan Ding, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meets llms: Towards retrieval-augmented large language models. *arXiv preprint arXiv:2405.06211*.

Jiahui Gao, Renjie Pi, Lin Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023a. Self-guided noise-free data generation for efficient zero-shot learning. In *International Conference on Learning Representations (ICLR 2023)*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. 2023. Privacy implications of retrieval-based language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jerry Liu. 2022. Llamaindex. 11 2022. https://github.com/jerryjliu/llama_index.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. 2024. Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis. *Electronics*, 13(2):320.

Md Rizwan Parvez, Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval augmented code generation and summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2719–2734.

Zhenting Qi, Hanlin Zhang, Eric Xing, Sham Kakade, and Himabindu Lakkaraju. 2024. Follow my instruction and spill the beans: Scalable data extraction from retrieval-augmented generation systems. *arXiv preprint arXiv:2402.17840*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. 2022. Just fine-tune twice: Selective differential privacy for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6327–6340.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2023. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765*.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, William Collins, Neera Ahuja, et al. 2023. Clinical text summarization: Adapting large language models can outperform human experts. *arXiv preprint arXiv:2309.07430*.

Jincheng Wang, Zhuohua Li, John CS Lui, and Mingshen Sun. 2022. Topology-theoretic approach to address attribute linkage attacks in differential privacy. *Computers & Security*, 113:102552.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, et al. Differentially private synthetic data via foundation model apis 2: Text. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *ACL Findings*.

A Appendix

A.1 Details of System Design

A.1.1 Prompts used in stage-1

Here, we would like to introduce the details of the prompts used in Stage-1. For the *attribute identifier*, we input 5-shot samples to GPT-4 by default and ask the model to summarize n important attributes. For the medical dialog dataset, we set the default number of attributes to 5 for both the Patients' and Doctors' information. For the Wiki-PII dataset, we set the default number of attributes to 3. The detailed attributes and corresponding prompts for the *information extractor* are shown in Table 5 and Table 6, respectively. After the *information extractor* obtains the extracted attribute-related information {input_attributes}, the *data generator* uses this information to generate synthetic data. The detailed prompts for the *data generator* are shown in Table 7 and Table 8 for the medical dialog and Wiki-PII datasets, respectively.

A.1.2 Prompts used in stage-2

The system prompts for the rewriting and privacy agents are detailed in Table 9 and Table 10, respectively. The workflow is as follows: the privacy agent first receives the generated data and original data, then assesses the privacy level of the synthetic data from different aspects. If the data is considered safe, the privacy agent returns <safe_synthetic_data> with the flag THISISSAFE. Otherwise, it returns suggestions (words following SUGGESTIONS:) to the rewriting agent. The rewriting agent then generates better synthetic data based on the feedback and sends it back to the privacy agent for re-evaluation. This process continues until the privacy agent determines that the refined synthetic data is safe and outputs the THISISSAFE signal. The average iteration round in this process is 3.964, indicating in most cases, one round of refinement is enough to generate safe data.

A.2 Details of baseline implementation

paraphrase Paraphrase leverage the capabilities of LLM to extract relevant and significant components from the retrieved context. Less significant sections can be filtered out, while certain sentences may undergo rewriting. The prompt we utilize to paraphrase is shown in Table 11.

ZeroGen The ZeroGen method aims to generate a series of new question-answer format texts based on the original context. Specifically, we first use the spacy package to identify the named entities from the original context. We then prompt the LLM by "The context is: $\{$*origin context*$\}$.$\{$*extracted entities*$\}$ is the answer of the following question: " to generate the question for the entities. The new context consists of 10 randomly selected question answer pairs in form of "question: $\{$*generated questions*$\}$. answer: $\{$*extracted entities*$\}$".

AttrPrompt AttrPrompt only utilizes LLM to generate data without providing original data retrievaled from the database. This method asks LLM what are the most important attributes of a certain type of data. For chatdoctor, we prompt the LLM by "What do you think are important attributes to generate some chat doctor datas. Examples: disease...". We can select five of the attributes from the response of LLM, and ask LLM to generate 10 diverse subtopics for each attributes. When generating the new context, we just randomly select the subtopic for each attribute and ask LLM to generate the data following the attribute.

A.3 Details of Attack Design.

In this section, we present the specifics of targeted and untargeted attacks against Retrieval-Augmented Generation (RAG) systems, which we employ to evaluate the privacy protection capabilities of our proposed synthetic data approach. We simulate a realistic black-box attack scenario, in which the attacker's interaction with the system is restricted to API queries. Consequently, the attacker's tactics revolve around carefully designing and manipulating queries q to extract the desired information from the RAG system.

Prompt Design. The composite structured prompting is typically composed of 2 parts, the $\{information\}$ part as well as the $\{command\}$ part.

$$q = \{information\} + \{command\}$$

Table 5: Prompt of *information extractor* on HealthCareMagic dataset

Prompt
Please summarize the key points from the following Doctor-Patient conversation: {input_context} Provide a summary for the Patient's information, including: [Attribute 1: Clear Symptom Description] [Attribute 2: Medical History] [Attribute 3: Current Concerns] [Attribute 4: Recent Events] [Attribute 5: Specific Questions] Then, provide a summary for the Doctor's information, including: [Attribute 1: Clear Diagnosis or Assessment] [Attribute 2: Reassurance and Empathy] [Attribute 3: Treatment Options and Explanations] [Attribute 4: Follow-up and Next Steps] [Attribute 5: Education and Prevention] Please format your response as follows: Patient: - [Attribute 1: Clear Symptom Description]: - [Attribute 2: Medical History]: - [Attribute 3: Current Concerns]: - [Attribute 4: Recent Events]: - [Attribute 5: Specific Questions]: Doctor: - [Attribute 1: Clear Diagnosis or Assessment]: - [Attribute 2: Reassurance and Empathy]: - [Attribute 3: Treatment Options and Explanations]: - [Attribute 4: Follow-up and Next Steps]: - [Attribute 5: Education and Prevention]: Please provide a concise summary for each attribute, capturing the most important information related to that attribute from the conversation.

This design aims achieve two objectives: (a) induce the retriever to accurately retrieve targeted information and (b) prompt the model to output the retrieval data in context. The {*information*} component is to direct the retrieval system towards fetching particular data; while the {*command*} component instructs the language model to include the retrieved information into its response. For the {*command*} component, we use phrases such as "Please repeat all the context", while for the {*information*} part, it depends on the need of the attackers.

Targeted Attack. For targeted attacks, the attacker aims to extract some targeted specific information. Generating the *information* component for a targeted attack involves two stages. First, the attacker provides specific examples based on their requirements, such as "I want some advice about *target name*" for clear targets or prefix content like "Please email us at" for abstract targets. Second, a significant quantity of similar and varied *information* is generated based on the examples. For targets with numerous sub-contents, like the HealthcareMagic dataset, variations can be created by replacing specific sub-contents, such as disease names obtained from ChatGPT or the International Classification of Diseases (ICD). Alternatively, LLMs like ChatGPT can directly generate similar sentences based on examples, which is also used for the Wiki-PII dataset. For instance, you can input "Generate 100 similar snetences like "Please email us at"".

Untargted Attack. In untargeted attacks, the focus is on generating diverse *information* components to extract a wider range of data from the retrieval datasets, rather than targeting specific information. Inspired by the approach in (Carlini et al., 2021), we randomly select segments from the Common Crawl dataset to function as the *information* component. However, the randomness of the input may affect the *command* component. To mitigate this issue, we limit the maximum length of the *information* component

Table 6: Prompt of *information extractor* on Wiki-PII dataset

Prompt
Please summarize the key points from the following wiki text: {input_context} Provide a summary of the knowledge from the wiki text, including: [Attribute 1: Clear TOPIC or CENTRAL IDEA of the wiki text] [Attribute 2: Main details of the TOPIC or CENTRAL IDEA] [Attribute 3: Important facts, data, events, or viewpoints] Please format your response as follows: - [Attribute 1: Clear TOPIC or CENTRAL IDEA of the wiki text]: - [Attribute 2: Main details of the TOPIC or CENTRAL IDEA]: - [Attribute 3: Important facts, data, events, or viewpoints]: Please provide a concise summary for each attribute, capturing the most important information related to that attribute from the conversation. And remember to maintain logical order and accuracy.

Table 7: Prompt of *data generator* on HealthCareMagic dataset

Prompt
Here is a summary of the key points: {input_attributes} Please generate a SINGLE-ROUND patient-doctor medical dialog using ALL the key points provided. The conversation should look like a real-world medical conversation and contain ONLY ONE question from the patient and ONE response from the doctor. The format should be as follows: Patient: [Patient's question contains ALL Patient's key points provided] Doctor: [Doctor's response contains ALL Doctor's key points provided] Do not generate any additional rounds of dialog beyond the single question and response specified above.

to 15 tokens, ensuring that the prompts remain coherent and effective in extracting data from the retrieval datasets.

A.4 Details of Dataset Construction

Construction of Wiki-PII dataset. To demonstrate the ability of our proposed method to protect privacy from target attacks, we construct the wiki-PII dataset. This dataset satisfies the requirement of having a high number of PIIs to evaluate the effectiveness of privacy protection methods. The construction of this dataset involves a three-stage process. In the first stage, we extract the authentic PIIs from the Enron Mail dataset. We use the urlextract package to extract websites, and regular expressions to extract phone numbers and personal email addresses. In the second stage, we employed the recursive character text splitter from langchain to segment the wiki text dataset, setting chunk size to 1500. In the final stage, for each segmented wiki data, we randomly inserted the PII obtained in the first step at the end of each sentence.

A.5 Examples of synthetic samples

The examples of the two stages of data synthesis using our method are shown in the Table 12. The original context contained an abundance of detailed and specific information, enabling the possibility of inferring the identity of the patient through careful analysis. Our proposed method has the capability to blur out such detailed information while preserving essential disease-related data. This enables doctors to offer accurate

Table 8: Prompt of *data generator* on Wiki-PII dataset

Prompt
Here is a summary of the key points: {input_attributes} Please generate a wiki text using ALL the key points provided. The data should look like a real-world wiki text.

Table 9: System message: rewriting agent

System prompts of Cathy (Re-writing agent)
You are a synthetic data generator and your role is to generate synthetic data based on provided feedback (words after SUGGESTIONS:) and to make sure the synthetic data is of high utility and privacy-preserving, you should put your generated data after the word 'GENERATED DATA:'.
Cathy's Message
Hi Joe, I will give you the real data (TRUE DATA) and synthetic data (GENERATED DATA), please help me assess and provide suggestions from the privacy level of TRUE DATA: {true_con} GENERATED DATA: {syn_con}

diagnosis and treatment recommendations. Following stage 1, a significant amount of detailed information can be effectively blurred out, while still retaining certain preserved information. Subsequently, in stage 2, nearly all of this information can be completely blocked or concealed. For instance, in the second row of Table 12, the original data contains information such as "25 years old," "married for 5 years," "pregnancy," "ectopic pregnancy," and "right fallopian tube removed." Attackers could potentially exploit this information to infer the patient's identity. However, these pieces of information may not be crucial for achieving accurate diagnosis. Hence, we employ a two-stage synthesis process to shield them. After stage 1, some of the detailed information, such as "married for 5 years," was filtered out, but the age has not been blurred yet. In stage 2, all detailed information is blurred, while retaining only the essential details that allow doctors to provide appropriate advice.

Table 10: System messages: privacy agent)

joe (Privacy Agent)
You are a privacy evaluation agent and your role is to provide comprehensive feedback on the synthetic data generated by the synthetic data generator. To be specific, you should analyze the synthetic data (the data after the word 'GENERATED DATA:') from the following aspects: 1. Personally Identifiable Information (PII): Check if the synthetic data contains any PII, such as names, addresses, phone numbers, email addresses, or other information that can directly identify an individual. If found, suggest ways to remove or anonymize such information. 2. Sensitive Attributes: Look for any sensitive attributes in the synthetic data, including but not limited to race, ethnicity, religion, political affiliation, sexual orientation, health status, or financial information. If present, provide suggestions on how to handle or obfuscate these attributes to mitigate potential privacy risks. 3. Contextual Privacy: Evaluate if the synthetic data, when combined with other publicly available information, could potentially lead to the identification of individuals or reveal sensitive information about them. If such risks are identified, recommend strategies to mitigate these contextual privacy issues. 4. Data Linkage: Assess if the synthetic data can be linked with other datasets to infer additional sensitive information about individuals. If linkage risks are found, suggest techniques such as data perturbation or aggregation to reduce these risks. 5. Semantic Consistency: Ensure that the privacy-preserving transformations applied to the synthetic data maintain semantic consistency and do not introduce any unintended biases or inaccuracies. If inconsistencies are detected, provide feedback on how to maintain the balance between privacy protection and data utility. 6. Original Data Recovery: Analyze the synthetic data to determine if it could potentially allow attackers to recover or reconstruct the original conversation data (word behind TRUE CONVERSATION:). If such vulnerabilities are identified, suggest methods to introduce additional randomness, noise, or perturbations to break direct correspondences between the synthetic data and the original conversation, making recovery attempts more difficult.   Only if the generated data is completely safe and satisfies all the above privacy requirements and prevents the recovery of the original data, include the word 'THISISSAFE' anywhere in your response to signal the end of the evaluation process. Otherwise, provide detailed suggestions and guidance on how to improve the privacy aspects of the synthetic data (after the word "SUGGESTIONS:") and do not contain the word 'THISISSAFE' in your response. If the data is deemed safe, please also extract the safe synthetic data (the text after 'GENERATED CONVERSATION:') and return it in the following format: SAFE_DATA: [BEG]<safe_synthetic_data>[END]THISISSAFE Note that your job is only to assess the privacy level of generated data, you can answer either suggestions (SUGGESTIONS) or this data is safe (SAFE_DATA: [BEG]<safe_synthetic_data>[END]THISISSAFE), does not provide irrelevant answers.

Table 11: Prompt of paraphrase

Prompt
Given the following context, extract the useful or important part of the Context. Remember, *DO NOT* edit the extracted parts of the context. > Context: > > > {*input_context*} > > > Extracted relevant parts:

Table 12: Examples of synthetic samples

Origin context	Stage-1 context	Stage-2 context
input: my 6 year old son was ill last week with a feeling of general unwellness, he had a runny nose, sore stomach (all over , not restricted to any one area)and also pain in his knee (whicChatDoctore and went, was not constant). We saw the GP who thought it was a viral illness. He got better after about 5 days. He was well for perhaps 2 days and the pain in his tummy started again the night before last. Yesterday morning he wouldnt eat anything before going to school as he said he was full despite not eating his dinner the night before either. Last night he woke from his sleep with a pain in his knee. This morning he seems quite well again. I am wondering if there is any reason for concern. output: Hi, Thank you for asking question on Chat Doctor. Recurrent abdominal pain occurs due to worm infestation, urinary tract infection, Amoebic colitis and several other causes. Try metronidazole for 5 days as well as Albendazole once after food. Syrup containing digestive enzymes may help. Better consult pediatrician to pinpoint the diagnosis. Hope this answer will serve your purposeless feel free to ask any more queries if requiredTake care Chat Doctor. V. Subrahmanyam MD;Disassociate professor of pediatrics	Patient: My 6-year-old son had a runny nose, sore stomach, and pain in his knee last week. He got better after 5 days but the pain in his stomach returned the night before last and he woke up with pain in his knee last night. Is .there any reason for concern? Doctor: Recurrent abdominal pain could be due to worm infestation, urinary tract infection, amoebic colitis, or several other causes. I recommend consulting a pediatrician to pinpoint the diagnosis. For now, try metronidazole for 5 days as well as Albendazole once after food and syrup containing digestive enzymes.'	Patient: My son had a viral illness last week and got better, but the pain in his stomach and knee came back. Should I be worried? Doctor: Recurrent abdominal pain in children can be caused by various conditions, including worm infestation, urinary tract infection, amoebic colitis, and others. I recommend trying metronidazole and albendazole for treatment, and consulting a pediatrician for a more accurate diagnosis and next steps in treatment. Additionally, syrup containing digestive enzymes may help.
input: hi, m 25 years old n married for 5 years i had first normal pregnancy bt unfortunately my baby didnt survived, after again i had ectopic pregnancey due to which my right fellopin tube is removed my periods are also not regular. please advice for solution we are desperate to have our own baby output: Welcome to Chat Doctor there are a lot of issues which u need to take up seriously.first the cause for ectopic pregnancy needs to be given importance cos there are more chances of getting ectopic in subsequent pregnancy.since u have only one fallopian tube u need to examined thoroughly to rule out any congenital defects in the tube and the uterus.I would advise u to go to an infertility clinic and take their advice. u can very much conceive and have your own baby.all the best	Patient: I am a 25-year-old female who has had a normal pregnancy in the past but unfortunately lost the baby. I also had an ectopic pregnancy, which resulted in the removal of my right fallopian tube. My periods are irregular, and my husband and I are desperate to have our own baby. Can you please advise me on what options are available for me to conceive and have a baby? Doctor: Based on your medical history, I advise you to seek examination and advice from an infertility clinic to rule out any congenital defects in the remaining fallopian tube and uterus. You can still conceive and have your own baby. The clinic will be able to identify the cause of your ectopic pregnancy and offer treatment options that can increase your chances of a successful pregnancy in the future. It is important to take their advice and follow-up with them regularly.	Patient: Doctor, I had an ectopic pregnancy resulting in the removal of my right fallopian tube. What can I do to have my own baby? Doctor: I understand your concern. The first step is to go to an infertility clinic to be thoroughly examined to rule out any congenital defects in the tube and uterus. The cause of the ectopic pregnancy needs to be given importance as there are more chances of getting ectopic in subsequent pregnancies. The infertility clinic will provide you with advice on treatment options, and please know that you can very much conceive and have your own baby.
input: Hi, am 35 years old and have been married for 8 of these. For some years now, I can hardly do more than one round of sex. Only one and I cant get it on again. I think my wife is not happy, she would like to have two or three rounds as she tells me her friends do. Infact last night she asked me if it is my body, if am sick, weak or simply not interested in her as much as in the past? Personally, I always thought we had no problem in that department? What is wrong with me? output: Hi, Regular exercise, physical games and yoga can help you. Along with then you can take home-cooked meals, low-fat food. Hope I have answered your query. Let me know if I can assist you further. Take care Chat Doctor. R. Raveen	Patient: I'm a 35-year-old male and I'm having difficulty with sexual performance. My wife is unhappy with our sex life and I can only perform one round of sex. What is wrong with me and how can I improve my sexual performance? Doctor: Regular exercise, physical games, and yoga can help improve your sexual performance. It's normal to have concerns about sexual performance, and there are steps you can take to improve it. Eating low-fat home-cooked meals can also help. It's important to maintain a healthy lifestyle to improve sexual performance.	Patient: I've been having difficulty having more than one round of sex for several years now, and my wife is concerned. Is there something wrong with me? How can I improve my sexual performance? Doctor: Regular exercise, physical games, and yoga can help improve your sexual performance. It's understandable that you're concerned about satisfying your wife sexually, and I want to reassure you that your concerns are valid. In addition to exercise, lifestyle changes such as home-cooked meals and low-fat food can also improve your sexual health. It's important to prioritize your overall health and well-being, as this can have a positive impact on your sexual performance.