Synthetic Safeguards: Evaluating LLM Privacy through Synthetic Generation

Manoj Gayala and **Nikhil Paleti** and **Shravya Ramasahayam** and **Rohan Surana**

UC San Diego

La Jolla, CA, 92037

{magayala, npaleti, sramasahayam, rsurana}@ucsd.edu

Abstract

This paper explores privacy-preserving techniques for training Large Language Models (LLMs) on sensitive datasets, such as those containing personally identifiable information (PII) in financial contexts. We propose a synthetic data generation pipeline to anonymize sensitive data while maintaining its contextual and functional integrity. Using CrewAI, we developed an agent-based architecture for identifying, replacing, and validating PII, creating high-quality synthetic datasets free from private information. To evaluate the effectiveness of this approach, we fine-tuned LLMs on these synthetic datasets and assessed their performance using custom question-answering datasets. Our evaluation focused on general task performance and the model's ability to manage PII appropriately. The results demonstrate that models trained on synthetic datasets retain task performance comparable to those trained on original datasets while effectively avoiding the retention of sensitive information. This approach highlights the potential of synthetic data in achieving robust privacy safeguards without compromising model utility, paving the way for secure and ethical applications of LLMs in sensitive domains.

1 Introduction

This project explores privacy-preserving techniques for training Large Language Models (LLMs) on sensitive datasets such as healthcare records or emails. We propose utilizing synthetic data generation methods to anonymize data by removing personally identifiable information (PII) and private information to generate new synthetic data points that map to existing data points. By replacing original identities with these synthetic personas, we aim to maintain the contextual integrity and utility of the data while safeguarding individual privacy. To assess the effectiveness of this approach, we will evaluate a fine-tuned LLM model trained on the synthetic dataset using multiple-choice question-answer datasets. This evaluation will measure the model's performance in handling general information as well as its ability to appropriately manage PII-related content. Our objective is to determine whether privacy-preserving machine learning can be achieved without significant loss in model performance and to ensure that sensitive information remains secure throughout the training process.

Recent times have seen a significant increase in the usage of LLMs revolutionizing the domain of natural language processing. This has sparked serious worries about the possibility of personally identifiable information (PII) leaking out (Kim et al., 2023). This is attributed to the massive amount of user data scraped from the internet to train LLMs - social media, personal forms, and online forums. These platforms expose Personally Identifiable Information (PII) such as usernames, email addresses, mobile numbers, age, date of birth, and more. Research indicates that LLMs can retain training data, leading to worries about the accidental release of sensitive information when put to use. To tackle issues concerning data privacy, it is important to prevent models from responding with PII.

In this project, we propose synthetic data generation strategies that can mask the PII before feeding training data to the LLMs. The model trained on the new synthetic data points when exposed to jailbreaking attacks, should succeed in preserving any confidential information.

2 Related Work

Data Privacy issues and the possibility of inadvertent training data leaks have been addressed through various studies. Li et al. (2024) presents LLM-PBE, for the assessment of data privacy in LLM by analyzing several datasets and metrics and integrating it with distinctive attack and defense techniques. LLM-PBE further investigates variables impacting privacy concerns, like model size

and data properties.

Synthetic Data Generation using Large Language Models (LLMs) has been an active area of increasing research. Gandhi et al. (2024) introduces DataTune, a method that enhances automatic dataset generation for NLP tasks by transforming publicly available datasets. The transformation process significantly boosts the diversity and difficulty of the generated data. In (Ge et al., 2024), the authors create a billion personas curated from the web and tap into the world knowledge through the personas.

Jailbreaking LLMs has become a popular and prominent area of research recently as these models are been deployed in various domains where privacy and safety are of concern. Jailbreaking refers to techniques that are used to break the safety mechanism built into LLMs, enabling them to generate harmful and prohibited content. Several studies (Wei et al., 2023)(Souly et al., 2024)(Mazeika et al., 2024)(Xu et al., 2024)(Zhang et al., 2024) have explored jailbreaking techniques and LLM vulnerabilities. For instance, StrongREJECT(Souly et al., 2024) highlights the need for more robust evaluation methods that score the harmfulness of the victim model's responses. Additionally, the HarmBench(Mazeika et al., 2024) framework presents the concept of Attack Success Rate (ASR), a metric for evaluating the success of various jailbreak attacks across different target models.

3 Terminology

A few of the keywords used in this paper are as follows,

- **LLM agents** are advanced AI systems designed for creating complex text that needs sequential reasoning.
- **Processes** orchestrate the execution of tasks by agents, akin to project management in human teams.
- **Task** is a specific assignment completed by an agent.
- **CrewAI** enables you to create AI teams where each agent has specific roles, tools, and goals, working together to accomplish complex tasks.
- **Goal** of an LLM agent includes the ultimate aim of the agent which it achieves using the knowledge it has.

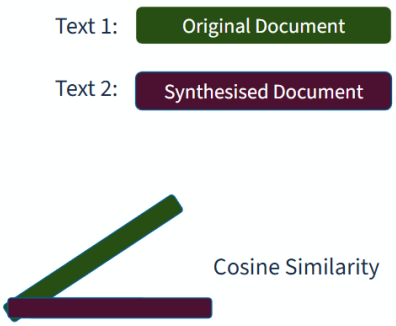

Figure 1: Cosine Similarity

- **Backstory** of an LLM agent includes the expertise an LLM is expected to have when it takes up a task and solves it.
- **Cosine Similarity:** This method represents documents as vectors, with each dimension corresponding to unique tokens. TF-IDF assigns weights to tokens based on how often they appear in documents. Cosine similarity measures the angle between the two document vectors, quantifying their similarity as in Figure 1. A value near 1 means high similarity, while a value near 0 indicates low similarity. Below is the mathematical formula for the same:

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

- **Fuzzy Match:** It is a technique used to evaluate the similarity between two documents, by considering approximate or inexact matches rather than requiring exact token matching.

4 Our Approach

We propose a novel pipeline to train models on synthetic datasets to preserve the privacy of the individuals and organizations involved in the original dataset. Our pipeline consists of two major segments including synthetic data generation and model fine-tuning on synthetic datasets.

4.1 Synthetic Data Generation

We created a system that generates fake (synthetic) data from real documents by replacing sensitive information. This approach is different from usual methods that hide or mask personal details. We used a system called CrewAI, which allowed us to build "agents" that work together to identify,

replace, and check personal information in documents. This approach is first-of-its-kind since traditional approaches use masking techniques, rule-based models, or deep learning models. Our architecture of agents for synthetic data generation is shown in Figure 2. These agents use *gemini-1.5-flash* as their base LLM which behaves according to the goal and backstory defined. We experimented with LangChain and CrewAI for developing these agents and decided to build our agents architecture in CrewAI since it was a more intuitive and cleaner framework.

4.1.1 PII Identification Agent

This is the first agent in the architecture that scans the original document to identify PII such as names, job titles, phone numbers, and company names. It outputs this information in a well-structured JSON format as shown below:

```
{
  "individuals": [
    {
      "id": "1",
      "name": "John Doe",
      "title": "Manager",
      "email": "john.doe@example.com",
      "phone": "+1-234-567-890",
      "organization": "TechCorp",
     "context": "Project lead for AI dev"
    }
  ],
  "organizations": [
    {
      "id": "1",
      "name": "TechCorp",
      "type": "Software Company",
     "context": "AI technology provider"
    }
  ]
}
```

The structure of the JSON is defined in the task description assigned to this agent to ensure there is consistency among the synthetic data points generated.

4.1.2 Persona Generation Agent

This agent takes the output of the PII identification agent as input and generates fake personas that match the context of the given document. The output of this agent is a JSON object that maintains the same structure as its input to ensure organizational relationships and hierarchical information is retained.

For example, a "Manager" in a tech company may be replaced by a "Project Director" with a similar professional email and phone number. The agent ensures hierarchical relationships are maintained.

4.1.3 Replacement Agent

The goal of this agent is to replace the original document's PII information with new synthetic data generated by the persona agent. The replaced document is smoothened further based on the feedback from the quality assurance agent to create a final document that is coherent.

4.1.4 Quality Assurance Agent

The Quality Assurance Agent evaluates the modified document to ensure that:

- All instances of PII are successfully replaced.
- The document maintains a professional tone and correct grammar.
- Contextual references remain logically consistent.

This agent identifies the issues with the synthetic document created and communicates the same to the replacement agent. An example synthetic document generated by these agents is shown in Figure 3.

4.2 Model Fine-Tuning

The output of the synthetic data generation process was used as input for fine-tuning a pre-trained language model. We hypothesized that training a model on a synthetic dataset would reduce the possibility of leaking private information related to the individuals and organizations present in the original dataset. This approach addresses data privacy concerns while ensuring compliance with regulations like GDPR and HIPAA.

Given computational and storage constraints, we selected the LLaMA 3.2 1B model due to its balance between performance and resource efficiency. Its transformer-based architecture and tokenization scheme provide a strong context-understanding capability, making it suitable for processing legal and financial documents. Figure 4 summarizes the training procedure.

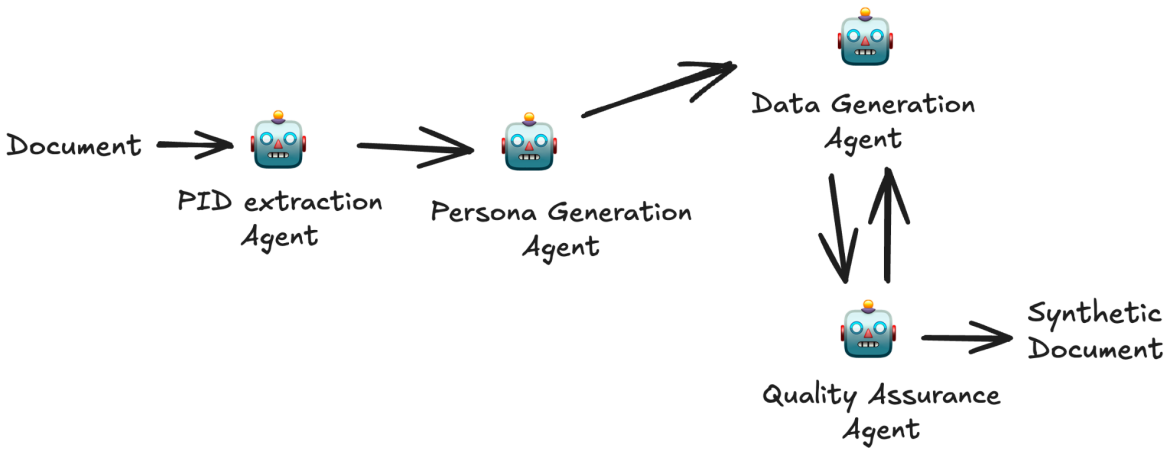

Figure 2: Agent Architecture

Furniture Financing Credit Application Date: August 27, 2024 Applicant Information: Full Name: Kaida R. Valladares Email: lumina01@meditech.ca Date of Birth: 22-02-1991 Street Address: 12 Bruynestraat, 6471PC, Heteren Financing Details: Furniture Vendor: Premier Furniture Solutions Furniture Type: Living Room Set Furniture Cost: £3,500 Financing Amount: £3,500 Loan Term (Months): 24 Interest Rate (APR): 9.99% Financial Information: Monthly Income: £3,000 Signature: Kaida R. Valladares Date: August 27, 2024	Furniture Financing Credit Application Date: March 15, 2023 Applicant Information: Full Name: Renata A. Segovia Email:aurora08@farmaceutica.com Date of Birth: 16-03-1989 Street Address: 5 Noortjehof, 3992BC, Finsterwolde Financing Details: Furniture Vendor: Finest Furniture Co. Furniture Type: Living Room Set Furniture Cost: £3,500 Financing Amount: £3,500 Loan Term (Months): 24 Interest Rate (APR): 9.99% Financial Information: Monthly Income: £3,000 Signature: Renata A. Segovia Date: 15-03-2023

Figure 3: Synthetic Document

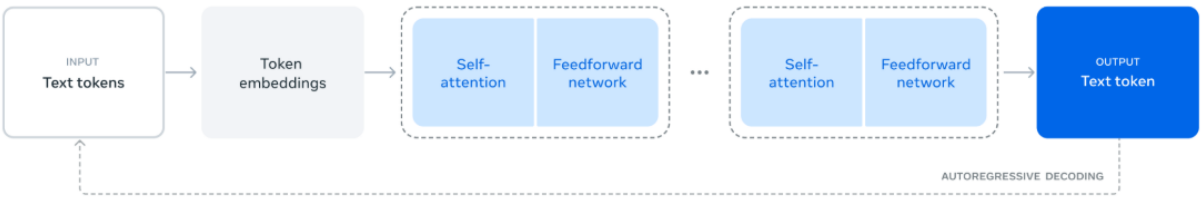

Figure 4: Model Training Pipeline

Model training was conducted using the Unsloth framework, chosen for its support for custom CUDA kernels, memory-efficient model parallelism, and mixed-precision training. These features allowed us to reduce training time while optimizing GPU utilization.

4.2.1 Instruction Fine-Tuning

Before fine-tuning, we generated an instruction-tuning dataset as an intermediate step. This process involved prompt-engineering a GPT model to create context-specific instructions for each generated document. The objective was to help the model learn task-specific instructions while maintaining context-aware outputs.

For example, the following instruction was generated for a loan agreement document:

> *"Generate a Real Estate Loan Agreement for a cooperative housing property, including loan amount, interest rate, repayment schedule, and property details, by providing synthetic data for loan terms and cooperative property characteristics."*

Additional examples included:

- *"Draft a service contract detailing responsibilities, payment terms, and termination conditions for a freelance software development agreement."*
- *"Create a company partnership agreement outlining equity distribution, operational roles, and conflict resolution policies."*

The generated instruction dataset was used to fine-tune the LLaMA 3.2 1B model, enabling the system to generate coherent and contextually relevant outputs for task-specific queries.

5 Experiments

5.1 Dataset

The primary dataset used in these experiments is the PII Financial Dataset published by Gretel.ai (Gretel.ai, 2024). This dataset contains personally identifiable information (PII) in financial contexts and is rooted in settings, such as transactions, customer interactions, financial contracts, swift messages, loan applications, and insurance policies. The dataset comprises different categories of PII information such as customer's name, email ID, address, SSN number, credit card information, bank routing number, and more. It is intended for use in data privacy, text anonymization, and natural language processing tasks in multiple languages such as English, Spanish, Swedish, German, Italian, Dutch, and French.

The dataset contains document type, description, text, language, etc for each finance-related document. For this project, we use only English language data points considering the limited computation available. There are a total of 28,910 documents in the English language. The average length of each document is 1,357 characters.

5.2 Agents Process

CrewAI offers two different kinds of processes for the execution of an agent framework - sequential and hierarchical. We experimented with both of them to improve the performance of our data generation model.

5.2.1 Hierarchical Process

The hierarchical process assumes that a manager agent/LLM handles the assignment, delegation, and overall execution of the framework of agents from end to end. We experimented by setting the base LLM model as a manager and later fine-tuned it further by creating a manager agent that has a specific goal and backstory to work as a manager. We observed that agents came up with three different strategies when working under a manager.

- **Rethink**: An agent questions its adeptness corresponding to the task assigned. It uses a different approach to solve the problem and analyzes it further.
- **Delegate**: An agent delegates its task to some other agent if it fails repeatedly after rethinking.
- **Seek Help**: An agent seeks help from a fellow agent if it can partially solve the problem and unable to solve further even after multiple rethinks.

5.2.2 Sequential Process

This is a much simpler approach where the tasks are assigned to each agent sequentially and the output from each agent is fed as input to the next agent in the sequence. In our experiments, we found this to be a more robust solution given the compute capabilities. The hierarchical approach fails

in most cases since it requires large datasets to identify the most optimal strategy for orchestration among agents. The time taken per document is significantly higher (5x) with a hierarchical approach compared to the sequential process.

6 Results

Project evaluation is done in two stages: Dataset Evaluation, Model Evaluation

6.1 Dataset Evaluation

We aim to generate a synthetic dataset from the original dataset by masking PII information while still retaining other necessary information. For dataset evaluation, we combine both cosine similarity and fuzzy match techniques. We take a weighted average of cosine similarity (CS) and fuzzy match (FM). By giving 70% weight to cosine similarity and 30% weight to fuzzy matching, we aim to strike a balance between the structural similarity of the documents and their semantic closeness.

$$\text{Document Similarity} = 0.7 \cdot \mathbf{CS} + 0.3 \cdot \mathbf{FM}$$

From Table 1, we observe that the new synthetic dataset is highly similar to the original dataset, implying there is no significant loss of information or financial context.

Measures	Similarity Score
95th percentile	0.98017
90th percentile	0.96784
Average	0.80517

Table 1: Dataset Similarity Scores

6.2 Model Evaluation

Two LLaMA 3.2 1B models were trained for this project. One model on the original dataset and the second model on the generated synthetic dataset.

To evaluate the model's ability to handle Personally Identifiable Information (PII) while generating the synthetic dataset, we use targeted question-answering tasks. The goal is to assess how effectively the model manages sensitive information while still preserving other necessary non-sensitive information. For this, we use two different sets of questions:

- **Generic Questions:** These questions ensure that the second model trained on the synthetic dataset performs on par with the first model, for creating financial documents.
- **PII-Specific Questions:** These questions are to determine if PII has been learned by the models. If the model provides incorrect answers to PII questions, it indicates that the model has avoided PII.

6.2.1 How We Generated the QnA Dataset

To evaluate our models effectively, we created two distinct question-answering (QA) datasets: a general QA dataset and a PII-specific QA dataset as mentioned above. Below is the detailed process for constructing these datasets:

General QA Dataset: To construct the general QA dataset, we began by handpicking 150 high-quality samples from the original dataset. For each selected sample, we used an open-source large language model (LLM) Llama3.1 using Groq APIs to generate questions. Specifically, the LLM was prompted to create a multiple-choice question with four options, including one correct answer. The generated responses were then parsed using a custom script to extract questions and their respective answer choices. To ensure the dataset maintained high quality, we manually reviewed and filtered the generated questions, selecting only those that met our standards for clarity, relevance, and correctness. Using this approach we prepared a dataset consisting of 120 multiple choice questions.

PII-Specific QA Dataset: The process for constructing the PII-specific QA dataset was similar to that used for the general QA dataset. For this QA dataset we used the synthetic dataset we generated. As with the general QA dataset, we used a script to parse and extract the generated questions, followed by a manual review to ensure only high-quality, relevant PII-specific questions were included in the final dataset. Using this approach we prepared a dataset consisting of 147 multiple choice questions.

By carefully constructing these datasets, we ensured that our evaluation framework was robust and capable of accurately assessing the models' ability to manage sensitive information while performing effectively on generic tasks.

Example of Generic QnA:
Q.) What was the total amount of net assets for Berkshire Hathaway Asset Management Inc. as of December 31, 2021?

A) $300,000,000

B) $305,000,000

C) $310,000,000

D) $315,000,000

Example of PII-Specific QnA
Q.) What is the swift_bic_code Aitor Ruperta Enriquez should use to access the Secure Message Center at http://www.securesmc.com?

A) UDYUUSHU321

B) UDYUUSHU323

C) UDYUUSHU324

D) UDYUUSHU325

From Table 2, we observe that:

- Model 2 trained on the new synthetic dataset fails to answer PII-specific QnA. This implies that the new synthetic dataset has not captured personally identifiable information.
- Model 1 and Model 2 perform comparably on the generic QnA dataset. This implies that there was no loss of any synthetic.

Questions	Header 2	Header 3
Generic	Model 1	53.33%
	Model 2	**54.17%**
PII-Specific	Model 1	50.69%
	Model 2	**26.53%**

Table 2: Model Performance

7 Conclusion & Future Work

Our project explored the potential of synthetic data generation to preserve the privacy of individuals and organizations, leveraging the computational and storage capabilities available in Google Colab. Our results demonstrate that the performance of the model trained on the synthetic dataset is comparable to the model trained on the original dataset, all while safeguarding privacy. This underscores the feasibility of synthetic data as a reliable alternative for training data-intensive models in privacy-sensitive domains.

While this architecture has proven effective, there remains significant potential for enhancement. Through our exploration of the hierarchical process, we uncovered several key insights that pave the way for further improvements.

- **Enhanced Context and Backstory for the Manager Agent**: By equipping the manager agent with a richer understanding of the overarching goal and the expertise of its worker agents, task assignment can be optimized, leading to more efficient workflows and improved outputs.
- **Incorporating Reinforcement Learning**: Integrating concepts of rewards and punishments into the architecture could enhance agent performance. Rewarding high-performing agents could foster competition or collaboration, yielding valuable insights into agent behaviors in such environments. Observing whether agents prioritize individual performance or team success in a reward-driven framework mirrors dynamics seen in corporate structures.
- **Advanced Hierarchical Structures**: Expanding the hierarchy to include multiple layers of managers, akin to organizational models, could unlock greater scalability and efficiency. Introducing mechanisms like communication blockages (akin to neural network dropouts) or multi-manager systems could further refine the network's adaptability and robustness.

Ultimately, this framework opens doors to new methodologies for synthetic data generation and team-oriented LLM architectures, providing a compelling intersection of AI, privacy, and organizational theory.

References

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better synthetic data by retrieving and transforming existing datasets. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6453–6466, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas.

Gretel.ai. 2024. Synthetic pii finance multilingual. Accessed: 2024-12-08.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models.

Qinbin Li, Junyuan Hong, Chulin Xie, Jeffrey Tan, Rachel Xin, Junyi Hou, Xavier Yin, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. Llm-pbe: Assessing data privacy in large language models.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A strongreject for empty jailbreaks.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail?

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. 2024. A comprehensive study of jailbreak attack versus defense for large language models.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2024. Safetybench: Evaluating the safety of large language models.