CNT 5410: Computer and Network Security

Mid-Semester Report: SafePrompt: A Privacy-Preserving Framework for PII Anonymization in LLM Interactions

Abdul Kalam Azad Shaik
(Point of Contact)
shaik.abdulkalam@ufl.edu

Baba Sai abhishek kothari
B.kothari@ufl.edu

Neil Rajeev John
neiljohn@ufl.edu

Aniruddh Atrey
aniruddh.atrey@ufl.edu

November 16, 2024

1 Introduction

The increasing use of Large Language Models (LLMs), such as ChatGPT, Claude.ai, and Gemini, has transformed how we interact with artificial intelligence across various sectors. These models, capable of sophisticated language understanding and generation, support diverse applications, from customer service to content creation. However, this widespread adoption introduces critical privacy concerns: users may inadvertently share sensitive Personally Identifiable Information (PII), and LLMs can memorize this data, making it vulnerable to extraction by adversarial attacks [5].

Such privacy risks are particularly concerning under data protection regulations like the General Data Protection Regulation (GDPR), which impose strict standards for safeguarding personal information. As LLMs become more integrated into everyday applications, finding solutions to protect user data without hindering model performance is crucial for building and maintaining public trust.

Our project proposes **SafePrompt**, a privacy-preserving framework designed to anonymize PII during interactions with LLMs. Acting as a protective layer, SafePrompt masks sensitive information in real-time before it is processed by the model and restores it afterward as needed. This approach mitigates privacy risks without compromising the quality of LLM outputs, providing a practical solution that addresses both user-side and model-side data exposure.

2 Background & Related Work

Significant advancements in de-identification have been made, especially in fields like healthcare, where protecting sensitive information such as patient identifiers and medical records is essential. Studies have developed various architectures to address this, such as the deep learning framework by Khin et al., which combined Bi-LSTMs and contextual embeddings to mask Protected Health Information (PHI) in EHRs, achieving state-of-the-art results [4]. Similarly, Johnson et al. demonstrated how fine-tuning BERT for de-identification tasks yielded high precision in detecting identifiers like names and dates in medical records [3]. While these methods are effective for static datasets, their adaptability to broader applications remains limited.

Beyond healthcare, Jensen et al. developed the JOBSTACK corpus for anonymizing job postings, focusing on privacy-sensitive entities like names, professions, and contact information. Their evaluation of Bi-LSTM and transformer models demonstrated the superior performance of pre-trained transformers in detecting sensitive entities [6]. These studies highlight the importance of tailoring de-identification methods to specific domains while emphasizing the limitations in addressing diverse and dynamic text sources.

Recent work by Zhang et al. explored data anonymization for LLM fine-tuning, leveraging Google's Data Loss Prevention (DLP) API to mask sensitive information in business call transcripts [2]. Their method integrated regex-based enhancements and context-aware anonymization, preserving semantic integrity during masking. This approach enabled fine-tuning of DialpadGPT, which achieved superior performance compared to models like GPT-3.5 and GPT-4 on tasks such as summarization and action item generation. While effective, their method faced challenges in generalizing to new domains due to its reliance on domain-specific taxonomies and tools. Adapting such approaches to different datasets and contexts requires substantial customization, limiting their broader applicability.

Another critical aspect of privacy preservation involves addressing information leakage risks during and after model training. Differentially Private Stochastic Gradient Descent (DPSGD) mitigates data retention risks by adding noise to training but introduces computational overhead and can affect accuracy [1]. Yu et al. proposed parameter-efficient fine-tuning techniques, balancing privacy and utility for large-scale models like RoBERTa [7]. Their approach highlights a growing need for privacy-focused techniques that address retention risks while minimizing performance trade-offs.

In response to these challenges, our project proposes SafePrompt, a novel framework designed to provide PII anonymization for LLMs. SafePrompt uses a fine-tuned Named Entity Recognition (NER) model to identify PII in user inputs, which is then masked using context-preserving placeholders before being processed by the LLM. This ensures that raw sensitive data is neither exposed to nor retained by the model. By replacing placeholders with contextually appropriate data (e.g., random but relevant names or locations), SafePrompt minimizes contextual loss, allowing LLMs to generate coherent and accurate responses.

The proposed framework addresses both user-side privacy concerns by masking sensitive inputs, and model-side risks by preventing LLMs from retaining sensitive data. The result is a scalable and effective privacy-preserving solution for LLMs that balances the need for robust performance with stringent privacy requirements across diverse applications.

3 Approach

SafePrompt is a privacy-preserving framework that addresses the risks of exposing sensitive Personally Identifiable Information (PII) during interactions with Large Language Models (LLMs). The framework operates through three key stages: PII Detection, Anonymization and Masking, and Demasking.

First, a fine-tuned Named Entity Recognition (NER) model is employed for **PII Detection** to identify sensitive entities in user inputs. Next, these entities are replaced with context-preserving placeholders during the **Anonymization and Masking** stage. These placeholders retain the semantic structure of the input, allowing the LLM to generate meaningful and coherent outputs without processing raw sensitive data.

Once the LLM generates its response, the framework performs **Demasking**, dynamically replacing the placeholders with their original values to produce a complete, contextually accurate output. By handling sensitive data this way, SafePrompt mitigates both user-side and model-side privacy risks.

SafePrompt is designed to be modular and adaptable, enabling its integration across diverse applications that require privacy compliance. Its approach ensures a balance between protecting user data and maintaining the utility of LLM outputs.

4 Preliminary Work & Results

This section outlines the foundational work performed to fine-tune and validate Named Entity Recognition (NER) models for detecting Personally Identifiable Information (PII) using synthetic datasets. Key efforts included data preparation, hyperparameter selection, model fine-tuning, and evaluation on independent test datasets. By systematically experimenting with different setups and consistently tracking performance metrics such as precision, recall, and F1-score, we established a robust framework for assessing PII detection capabilities, setting the stage for final evaluations on unseen test data.

4.1 Data Preparation

We began by selecting two distinct datasets, **400k** and **200k** from ai4privacy org on Huggingface, for training. The **400k dataset**[1], the largest open dataset for privacy masking, is designed to detect commonly occurring PII types such as ACCOUNTNUM, CREDITCARDNUMBER, and DATEOFBIRTH. With 406,896 examples, over 20 million tokens, and 2.3 million PII tokens across 17 labels, it provides a robust foundation for model training. In contrast, the **200k dataset**[2] offers a broader range of 56 PII categories, including nuanced entities such as MAC, BITCOINADDRESS, ETHEREUMADDRESS and PHONENUMBER, and even more personal attributes like HEIGHT, EYECOLOR ensuring stricter privacy protections. While smaller, with 209,000 examples and 649,000 PII tokens, this dataset is tailored for use cases demanding a higher level of privacy. Both datasets are multilingual, but we focused exclusively on English samples, resulting in 68,275 training samples and 17,046 validation samples for the 400k dataset, and 34,801 training samples and 8,700 validation samples for the 200k dataset.

The decision to train separate models on these datasets reflects their complementary strengths. The 400k-trained model is optimized for efficiency and prioritizes commonly encountered PII types, making it suitable for scenarios with moderate privacy needs. Meanwhile, the 200k-trained model provides comprehensive coverage of a broader range of attributes, addressing use cases that demand stricter privacy protections. Although merging the datasets was initially considered, differences in label schemes (17 vs. 56 labels) introduced risks of inconsistency, which could dilute model performance. Thus, the datasets were trained separately to maintain their respective strengths.

For testing, we selected the **beki/privy**[3] dataset, a synthetic test dataset with diverse formats such as JSON, SQL, and HTML, and 26 PII labels including PERSON, LOCATION, and CREDIT_CARD. This diversity offers realistic challenges for assessing model generalization. However, preprocessing revealed labeling inconsistencies; for example, the source text:

> INSERT INTO ‘prevent‘ VALUES (’CassandraRChildress@gustr.com’)

incorrectly labeled `CassandraRChildress@gustr.com` as O, even though it should have been classified as EMAIL. Despite these limitations, the dataset remains valuable for testing model robustness and adaptability.

Key preprocessing steps included tokenization, label alignment, and dataset splitting. The `bert-base-cased` and `roberta-base` tokenizers were used to split text into tokens and align them with BIO-encoded labels (B-, I-, O). Special tokens such as [CLS] and [SEP] were added for compatibility with transformer models, and padding tokens were assigned a label of -100 to exclude them from loss calculations. The 400k dataset utilized its predefined 80-20 train-validation split, while the 200k dataset was split into 90% training and 10% validation. Dynamic label mappings (e.g., `label_to_id` and `id_to_label`) were generated for each dataset, ensuring consistent integration during training and evaluation. Random tokenized samples for BERT[4] and RoBERTa[5] illustrate the preprocessing steps in detail. These efforts established a standardized foundation for fine-tuning and assessing PII detection models.

4.2 Fine-Tuning Models

Fine-tuning pre-trained language models for Named Entity Recognition (NER) is essential for adapting them to specific tasks like PII detection. For this project, we fine-tuned `bert-base-cased` and `roberta-base` models on the 400k and 200k datasets to address diverse privacy needs. BERT was chosen for its ability to effectively handle case-sensitive data, making it well-suited for identifying PII like names and emails. RoBERTa, on the other hand, excelled at capturing nuanced contextual relationships, offering robust performance for broader PII detection tasks. Both models were initialized with dynamically aligned tokenizers to ensure compatibility with their respective datasets.

The models were optimized with **PII Recall** as the primary metric, defined as the ability to detect PII tokens without missing any. This focus was critical to minimizing false negatives (PII tokens classified as O) and ensuring private information was not overlooked. A binary classification approach, classifying tokens as either PII or

[1] https://huggingface.co/datasets/ai4privacy/pii-masking-400k
[2] https://huggingface.co/datasets/ai4privacy/pii-masking-200k
[3] https://huggingface.co/datasets/beki/privy
[4] https://example.com/bert-tokenized-sample
[5] https://example.com/roberta-tokenized-sample

non-PII, reinforced this goal. While token-level precision, F1-score, and accuracy were also tracked as secondary metrics, **PII Recall** remained the main driver for model improvements.

To identify the best-fit configurations for each model, we employed **Bayesian optimization**, which efficiently explored the hyperparameter space compared to traditional grid search. The `bert-base-cased` model trained on the 400k dataset was initially tested across 15 configurations, allowing us to observe consistent performance trends. Based on these results, subsequent models (`bert-base-cased` on the 200k dataset and `roberta-base` on both 400k and 200k datasets) were tested across 10 configurations each. This approach balanced thorough experimentation with computational efficiency, ensuring robust performance across all datasets and models[6].

Hyperparameter tuning was conducted over key parameters: learning rates (`1e-5` to `5e-5`), batch sizes (8 and 16), weight decay (`1e-4` to `1e-2`), warmup steps (100 to 500), and dropout rates (0.1 to 0.3). The AdamW optimizer, configured with weight decay, was used to enhance convergence while mitigating overfitting. Linear and cosine schedulers were tested to dynamically adjust learning rates, with linear scheduling yielding more stable results.

Training strategies were refined for efficiency and performance. Gradient accumulation was initially set to 2 to simulate larger batch sizes but was reverted to 1 to optimize training speed without sacrificing performance. Mixed precision (`fp16`) training, while tested, was excluded from final experiments due to added complexity and negligible gains. Training was capped at five epochs for all configurations to maintain consistency. The `load_best_model_at_end=True` setting ensured that the best-performing model was saved automatically based on validation metrics, negating the need for early stopping.

Training progress and results were meticulously tracked using **Weights & Biases (W&B)**[7]. This facilitated detailed logging of metrics such as training and validation loss, hyperparameter sweeps, and visualizations for comparison. Intermediate checkpoints were saved to preserve the best-performing models for subsequent evaluation. These efforts ensured that the fine-tuned models were optimized for detecting PII effectively, balancing computational efficiency and robust performance.

Code Availability: All the source code is currently available on HyperGator and can be accessed at the path `/blue/cnt5410/shaik.abdulkalam/cnt5410`[8]. Due to an issue with HyperGator preventing me from pushing the code to GitHub, the repository will be updated at https://github.com/abdulkalam556/safe-prompt as soon as the issue is resolved. In the meantime, HyperGator remains the primary access point for the project code.

5 Next Steps

The immediate next steps involve selecting and evaluating the best models from the fine-tuned configurations. Evaluation metrics for all configurations have been stored in the output directory of the repository. Based on these metrics, the top three models for each configuration (`bert-400k`, `bert-200k`, `roberta-400k`, `roberta-200k`) will be selected and tested on the independent test dataset. This testing phase will produce various metrics, which will be analyzed to finalize the best-performing model for each configuration.

Using the predictions of the selected models on the test dataset, the next objective is to generate masked text. The source texts will be masked by replacing detected PII with placeholders. To ensure the masked texts remain contextually meaningful, the placeholders will be replaced with random but appropriate values. For example, a masked name label will be substituted with a random name, preserving the contextual integrity of the text.

These source texts and their contextually masked versions will then be processed by a chat-based LLaMA model through API calls. Outputs generated using the original source texts and the contextually masked texts will be compared using metrics such as ROUGE and BLEU to evaluate their similarity. Additionally, embedding similarity scores are being explored as a potential alternative to BLEU, as they may provide a more robust measure of the semantic consistency between the outputs.

[6] All configuration result plots can be viewed at https://github.com/abdulkalam556/safe-prompt-plots

[7] https://wandb.ai/site

[8] https://ood.rc.ufl.edu/pun/sys/dashboard/files/fs//blue/cnt5410/shaik.abdulkalam/cnt5410

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Shayna Gardiner, Tania Habib, Kevin Humphreys, Masha Azizi, Frederic Mailhot, Anne Paling, Preston Thomas, and Nathan Zhang. Data anonymization for privacy-preserving large language model fine-tuning on call transcripts. In *Proceedings of the Workshop on Computational Approaches to Language Data Pseudonymization (CALD-pseudo 2024)*, pages 64–75, 2024.

[3] Alistair EW Johnson, Lucas Bulgarelli, and Tom J Pollard. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 214–221, 2020.

[4] Kaung Khin, Philipp Burckhardt, and Rema Padman. A deep learning architecture for de-identification of patient notes: Implementation and evaluation. *arXiv preprint arXiv:1810.01570*, 2018.

[5] Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. Pii-compass: Guiding llm training data extraction prompts towards the target pii via grounding. *arXiv preprint arXiv:2407.02943*, 2024.

[6] Kristian Nørgaard Jensen, Mike Zhang, and Barbara Plank. De-identification of privacy-related entities in job postings. *arXiv e-prints*, pages arXiv–2105, 2021.

[7] Tianli Yu and Others. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.