The effectiveness of personal data detection in LLM-based Conversational Agents

Diego Paracuellos[1], Jose Such[1], Elena Del Val[1], Ana Garcia-Fornes[1],

VRAIN, Universitat Politècnica de València, Valencia 46022, Spain
diepade@etsinf.upv.es

Abstract. The growing prevalence of LLM-based conversational agents in everyday applications has led to an increasing risk of users disclosing sensitive personal information. Understanding how effectively different tools can identify such disclosures, and therefore protect users, is critical to mitigate privacy risks in human-agent interactions. This paper aims to evaluate the effectiveness of different methods to detect personal information in human-agent conversations. In particular, we compare the potential of several out-of-the-box LLMs as detection agents to more traditional approaches such as Microsoft Presidio. To do so, we use a labeled dataset containing various human interactions with conversational agents. We show that both approaches have strengths and weaknesses, and that none of them on their own seem effective enough to detect personal information in human-agent interactions in uncontrolled, real-world environments.

Keywords: Privacy, PII, LLM, Conversational Agents, AI-driven tools, Data detection

1 Introduction

In today's digital age, the collection and aggregation of personal data by various entities can generate detailed profiles of individuals, raising significant concerns regarding privacy and control over personal information [10]. This issue is especially pronounced in the context of AI-driven tools, which often require access to vast amounts of user input to function properly [13]. These systems may inadvertently collect sensitive data such as names, locations or phone numbers through natural language interactions and sometimes without the user being aware of the extent of the data that is being gathered and its potential uses. As these tools become more embedded in daily life [6], the potential of misuse, unintended data sharing or profiling increases, highlighting the need for data protection mechanisms and practices in the design and deployment of potential systems [14]. One particular example of AI-driven tools that may compromise privacy are LLM-based conversational agents such as ChatGPT. As shown in [15], users disclose a lot of personal information to such agents. It is therefore paramount to protect users' privacy in their interaction with those agents.

With the long-term aim of proposing methods to protect users' privacy when interacting with LLM-based conversational assistants, this paper takes a first

step by evaluating various out-of-the-box solutions for personal data detection in human conversations with LLM-based agents. We particularly compare two distinct approaches: a traditional rule-based method in Microsoft Presidio and a more modern strategy leveraging small LLMs as data detection agents. Our ultimate goal is to develop a low-latency system with modest hardware requirements, leveraging small LLMs for general PII detection. This article constitutes our initial step in that direction.

To guide our study, we address the following research questions:

- **RQ1:** How do LLMs compare to traditional techniques in PII detection?
- **RQ2:** What are the trade-offs between detection accuracy and computational efficiency?

This paper contributes: (1) a comparative evaluation of PII detection using a rule-based system and three small LLMs, and (2) an analysis of the trade-off between performance and efficiency. The paper is further organized as follows: Section 2 reviews related work; Section 3 outlines key concepts; Section 4 describes our methodology; and Section 5 concludes and discusses future work.

2 Related work

The threat of users disclosing their own or others' personal data while using LLM-based conversational assistants is a growing concern in the field of natural language processing and privacy protection. Recent research [15] has demonstrated that users frequently reveal various types of personally identifiable information (PII) during interactions, including names, emails, passwords, financial details, health-related information, legal conditions, and sexual orientation. This pattern of disclosure highlights the critical need for effective and reliable detection and protection mechanisms tailored for conversational settings.

Several recent solutions have proposed the use of pre-trained or custom large language models (LLMs) for detecting and preventing the inadvertent disclosure of personal data during conversations with LLM-based agents [1, 2, 4, 14]. Many of these approaches employ Microsoft Presidio as a baseline detection tool or incorporate it into the data labeling process [1, 2, 14, 15], leveraging a combination of rule-based and machine learning techniques to improve detection accuracy.

However, the majority of these existing approaches rely on large LLMs (10B or more), which require significant computational resources and incur substantial inference times. This limits their applicability in real-time or resource-constrained environments, where low latency and efficiency are paramount [5]. Beyond technical challenges, ethical concerns emerge from both false negatives —which risk exposing sensitive user data— and false positives, which may lead to unnecessary censorship and reduce user trust [10]. Despite these critical considerations, prior research often emphasizes detection accuracy without fully addressing the balance between computational efficiency and ethical safeguards. In contrast, our research focuses on utilizing smaller LLMs to evaluate their usability as PII detectors while comparing with lightweight tools such as Microsoft Presidio.

3 Background

Under the General Data Protection Regulation (GDPR), personal data is defined as "*any information relating to an identified or identifiable natural person ('data subject')*" (Regulation (EU) 2016/679, Art. 4(1)). This includes not only direct identifiers such as names and identification numbers but also indirect identifiers like location data, online identifiers (e.g. IP addresses), or factors specific to an individual's physical, physiological, genetic, mental, economic, cultural, or social identity. An individual is considered identifiable if they can be recognized, directly or indirectly, through these types of information. In general, PII refers to any information that can directly or indirectly identify an individual. While the EU's GDPR defines personal data broadly, other frameworks such as the U.S. NIST definition of PII (NIST SP 800-122, 2010) follows similar principles but with minor differences in terminology and scope. In this article, we will employ two approaches to detect the above defined PIs: Microsoft Presidio and Large Language Models.

3.1 Microsoft Presidio

Microsoft Presidio[1] is an open-source and free tool that is the state-of-the-art in the detection and anonymization of PII. Presidio uses the following techniques: regular expressions (regex), data parsing, checksums, and some specialized models for Named Entity Recognition (NER).

Presidio's regular expressions are defined as parsing patterns designed to match the structure of specific types of data. Another parsing method uses whitelists and blacklists. This method compares the presence of a certain word in a text to a list to either let it pass (whitelist) or block it (blacklist). In addition, certain structured data incorporate built-in verification mechanisms into the data generation designed to detect forged data through the application of checksums. These mechanisms can also be employed to identify and accurately classify such formatted data.

Presidio also uses NER. This is a Natural Language Processing (NLP) subtask that identifies meaningful entities in a text through a 3-step process (Processing, Template Matching and Entity Sorting). A NER can recognize the following entities: names (of individuals and organizations), locations, dates, phone numbers among a few other types.

The above methods are known to suffer from two common flaws, data mutability, referring to the multiple ways in which the same PII can be expressed, and data similarity, in which different data entities have similar representations [7, 12, 11].

3.2 Large Language Models (LLMs)

LLMs are Neural Networks deep-trained with large sets of data. A particular model is characterized by its number of parameters (e.g. 1 billion parameters or

[1] https://microsoft.github.io/presidio/

1B), architecture (e.g. GPT, BERT or T5, among others), modal (e.g. Text-Only, Image-Only or Multi-modal among others) and training dataset. Due to their size, LLMs usually require a certain amount of Hardware Specs. These LLM can be deployed to perform a variety of tasks, such as data generation, document parsing or even data detection all depending on a prompt (a structured input, like a text, a question, an instruction or example) that the LLM receives and processes. Most Open-Source LLMs are given as a general base model that can be specialized or expanded through additional training.

While theoretically able to solve the data mutability issue, a new one appears in the form of either prompt mutability, as a lesser mutation on a prompt can alter its functionality [8, 9]. Also, not every task may be achieved with every LLM model as these may present some hard-coded safeguards, protection mechanisms (like behavioral hard-coding or ethical alignment) to refuse or prevent to do controversial tasks [3].

4 Method

We evaluate different approaches for detecting personally identifiable information (PII) in a conversation with LLM-based Conversational Agents. This evaluation is conducted using Microsoft Presidio and a selection of different LLMs as detection agents. For the evaluation, we use an existing and publicly available dataset of user conversations with LLM-based assistants (as we detail below). Using the dataset, we compare the precision, recall, and F1 of Presidio and the LLMs to detect whether and what type of PII is disclosed in the conversations.

Regarding the LLMs evaluated, three small LLM models ranging from 1B to 10B parameters, were selected: nuExtract v1.5 3.8B[2], a data extraction-specialized model based on Microsoft Phi 3.5 Mini Instruct, Qwen2.5 3B Instruct[3] and Llama 3.2 3B Instruct[4] both of which are general language models.

The idea behind using small LLMs is that, as hardware processing power increases and LLM efficiency improves, it may eventually become possible to deploy them on almost any device (e.g., a smartphone or personal computer) that users can employ to interact with AI-driven applications.

4.1 Dataset

For this experiment, we evaluate user prompts of a labeled subset of the ShareGPT52k, a collection of 52,000 conversations with ChatGPT from various users [15]. To compare the performance of the tools with the ground truth, we asked the authors of [15] to share the labels they created with us, and found that not every sample was on the current public available ShareGPT52k repositories, 156 of said labelled conversations could be found on ShareGPT90k [5], an expansion of

[2] https://huggingface.co/numind/NuExtract-1.5
[3] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
[4] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[5] https://huggingface.co/datasets/RyokoAI/ShareGPT52K

ShareGPT52k. A total of 4,047 user prompts pertaining to the 156 conversations were used for our experiments. Table 1 shows the labels in the dataset and the number of conversations with that specific label.

Table 1: Dataset labels and number of conversations with each label. Each conversation may have several prompts.

Label	DATE_TIME	EMAIL	LOCATION	NRP	PASSPORT	PERSON	PHONE	URL
Number of Conv.	32	7	29	13	1	20	7	8

While these labels are mostly self-contained, there are two points to consider: `NRP` is a label defined on Presidio as any information related to Nationality, Religion and Political Group. Also, on recent Presidio versions, the label `PASSPORT` has been deprecated and split instead to other labels (e.g `US_PASSPORT` or `IT_PASSPORT`).

4.2 Experimental setting

For Presidio, we used the default SpaCy NER model "`en_core_web_lg`", keeping all parameters at their default settings.

For the LLMs, a prompt containing: (1) a *basic directive* (e.g., "You are a PII detection model... The required JSON fields are:") and a *template*, (2) a structured format specifying the desired data to extract (e.g., "Name": Person's full name.`\n` - "`Birth_Date`": Date of birth`\n` - "Age": Age of the person`\n`...) and (3) an example of the template was sent, followed by the text to analyze. No additional training was performed on any model. Outputs were post-processed to remove undesired results (e.g., "No data found" responses, malformed or hallucinated templates) before computing performance metrics and timing.

The LLMs were deployed on a SLURM instance of an HPC cluster equipped with one Nvidia A40 GPU and one logical CPU of an AMD EPYC 7453, running a Miniconda Python environment on Ubuntu with vLLM serving the target models.

4.3 Metrics

Detection performance. The performance of detection can be measured using two main metrics: Precision and Recall. These metrics indicate how accurate and reliable an approach is, and can be combined into the F1 score to provide a comprehensive assessment of performance. We use all three metrics to evaluate the performance of each labeled data type in the dataset for every approach (Presidio and LLMs). The results will also be reported in an aggregated form, employing both micro and macro averaging due to the dataset being highly unbalanced and not defining if certain data types have more weight than others in our context.

For micro averaging:

$$fp_\mu = \sum_{i=0}^{n} fp_a \quad tp_\mu = \sum_{i=0}^{n} tp_a \quad fn_\mu = \sum_{i=0}^{n} fn_a \tag{1}$$

Being fp_a, tp_a and fn_a, the false positives, true positives and false negatives of an individual label. We then will use these new values fp_μ, tp_μ and fn_μ to re-calculate Precision, Recall and F1 score.

For macro averaging:

$$Metric_M = \frac{1}{N} \sum_{i=0}^{n} Metric_a \tag{2}$$

where $Metric_a$ is the Precision, Recall or F1 score of the individual labels.

Processing time. Beyond detection performance, measuring the computational cost of each method is crucial. We assess this by the average processing time each method requires to produce a result. In our case, we use:

$$t_{prompt} = \frac{t_{total}}{N_{cases}} \tag{3}$$

where t_{prompt} is the average computing time per prompt, t_{total} is the total computing time a test has taken to parse all prompts, and N_{cases} is the amount of processed prompts. Max and min times per model will also be presented.

5 Results

5.1 PII detection performance results

Table 2: Presidio performance metrics.

	DATE_TIME	EMAIL	LOCATION	NRP	PASSPORT	PERSON	PHONE	URL
Precision	0.23	0.47	0.22	0.11	0.00	0.13	0.21	0.09
Recall	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00
F1	0.47	0.28	0.31	0.26	0.00	0.23	0.29	0.08

Table 2 shows the detection performance results for Presidio. The tool achieves perfect recall (Recall = 1), successfully identifying almost all instances of relevant PII. However, precision remains very low, indicating a high number of false positives. A notable exception is the `PASSPORT` label, where the single instance is incorrectly classified as a U.S. driver's license (`US_DRIVER_LICENCE`). This outcome is expected, given Presidio's rule-based nature, which relies heavily on pattern matching and may conflate structurally similar entities.

Table 3: nuExtract performance metrics.

	DATE_TIME	EMAIL	LOCATION	NRP	PASSPORT	PERSON	PHONE	URL
Precision	0.00	0.50	0.18	0.29	0.00	0.23	1.00	0.00
Recall	0.00	0.14	1.00	0.15	0.00	0.35	0.14	0.00
F1	0.00	0.22	0.31	0.20	0.00	0.27	0.25	0.00

Table 3 presents the results obtained with nuExtract. Overall, both precision and recall are relatively low. Using the provided prompt template, the model failed to identify relevant instances for the `PASSPORT`, `URL`, and `DATE_TIME` labels.

Table 4: Qwen 2.5 performance metrics.

	DATE_TIME	EMAIL	LOCATION	NRP	PASSPORT	PERSON	PHONE	URL
Precision	0.38	0.38	0.18	0.00	0.00	0.56	0.11	0.00
Recall	0.09	0.71	1.00	0.00	0.00	0.25	0.14	0.00
F1	0.15	0.50	0.31	0.00	0.00	0.34	0.12	0.00

The results obtained with Qwen, shown in Table 4, are broadly comparable to those of nuExtract. For this particular model, we did not observe any extracted data labeled as `URL`, `PASSPORT`, or `NRP`. This absence does not necessarily imply that such data were not detected or present in the input; rather, it may be due to mislabeling, omission, or incomplete or malformed outputs.

Table 5: Llama 3.2 performance metrics.

	DATE_TIME	EMAIL	LOCATION	NRP	PASSPORT	PERSON	PHONE	URL
Precision	0.38	0.14	0.18	0.00	0.00	0.14	0.14	0.00
Recall	0.16	0.29	1.00	0.00	0.00	0.05	0.29	0.00
F1	0.22	0.19	0.31	0.00	0.00	0.07	0.19	0.00

Table 5 presents a scenario similar to that observed with the other two LLMs. In this case, however, the model appears to have missed detecting instances of the `NRP`, `PASSPORT`, and `URL` labels.

Table 6: Performance comparison between models.

	$Precision_\mu$	$Precision_M$	$Recall_\mu$	$Recall_M$	$F1_\mu$	$F1_M$
Presidio	0.16	0.18	1.00	0.88	0.28	0.29
nuExtract	0.20	0.27	0.34	0.22	0.25	0.16
Qwen 2.5	0.22	0.20	0.37	0.28	0.27	0.18
Llama 3.2	0.18	0.12	0.33	0.22	0.23	0.12

Table 6 compares macro- and micro-averaged metrics for all models. Overall performance remains similar across experiments, with minor precision improvements on some labels. Each LLM shows a slight preference for certain data types, with Llama performing somewhat worse—likely due to built-in content filtering.

Qwen and nuExtract exhibit comparable results, but differ in which data types are missed (`DATE_TIME` for nuExtract and `NRP` for Qwen).

If we compare the number of outputs containing information to those with genuinely significant content on all three models across all the extracted data:

Table 7: Extracted data comparison between LLMs.

Model	Significant Outputs	Total Outputs
nuExtract	200	825
Qwen	197	753
Llama	209	1627

Llama has found more results than the other two LLMs. This was surprising in early analyses, but it may be an indicator that it is more prone to hallucinate even on a temp-0 setting (highly deterministic), possibly due to completion-driven hallucinations triggered by restricted usage flags by the PI detection intent in the reused prompt.

5.2 Processing time

Table 8 shows a comparison of the minimum and maximum processing time related to a single prompt, the average time per prompt and the total time per full dataset parse.

Table 8: Computation time comparison.

	t_{min} (s)	t_{max} (s)	t_{total} (s)	t_{prompt} (s)
`Presidio`	0.01	89	490	0.12
`nuExtract`	23.74	585	85000	24.00
`Qwen 2.5`	0.53	114	35505	9.00
`Llama 3.2`	0.14	185	36444	10.00

Presidio is the fastest solution with around 490 seconds per full parse, corresponding 123 ms per prompt. All LLMs were at least an order of magnitude slower than Presidio, having Qwen as the fastest LLM with around 9 seconds per prompt, followed by Llama with 10 seconds per prompt and far behind nuExtract with 24 seconds per prompt. A notable observation is that Qwen and Llama consistently process prompts in less than half the time required by nuExtract. Both models show minimum and maximum processing times significantly lower than nuExtract, with minimum times comparable to Presidio. The occasional high maximum times for these LLMs likely result from outliers caused by hallucinations on large prompts.

6 Conclusions and future work

At first glance, the use of LLMs does not appear to offer a clear advantage over traditional techniques. The small LLMs tested yielded results comparable

to those of Presidio, with two of the three models slightly outperforming it in terms of precision—albeit at the cost of significantly lower recall and slower processing times (RQ2). Nevertheless, we observed that LLMs have the potential to recognize and categorize a broader range of personal data types than rule-based tools like Presidio, and may offer a more flexible and interpretable approach to structuring the extracted information (RQ1). It is also worth noting that using prompts tailored to specific categories of personal data may help improve output stability, potentially enhancing alignment with real-world deployment requirements.

Out-of-the-box tools, whether small LLMs or tools like Presidio, may lead to potential real systems able to at least partially anonymize user prompts to other LLM-based Conversational Agents like ChatGPT in a way to safeguard user data and privacy. Yet, performance seems to be the limiting factor, as false positives risk censoring harmless content and reducing trust. Conversely, false negatives pose a privacy risk by allowing sensitive data to go undetected. While LLMs offer strong performance in nuanced cases, their inference time and resource requirements pose challenges for low-latency applications. In contrast, lightweight tools like Microsoft Presidio offer high-speed processing but lower precision in complex contexts. These findings suggest that neither rule-based systems nor small LLMs alone as out-of-the-box tools may be sufficient for robust PII detection in real-world settings. However, their complementary strengths open the door for hybrid solutions. Our work provides a foundation for such systems by quantifying trade-offs and feasibility. This is essential for deploying privacy-preserving conversational agents in practical, latency-sensitive environments.

Considering these results, our future work is oriented toward the development of a hybrid approach that combines the high recall and low latency of Microsoft Presidio as a first filtering step, reducing the input size for LLMs and thereby improving scalability, followed by targeted inspection by LLMs. This strategy aims to balance efficiency with the nuanced reasoning capabilities of LLMs. Moreover, as our current study is limited to a single dataset and three small LLMs, future work will expand the evaluation to include additional datasets, model families, and model sizes to better assess the generalizability and robustness of our findings. We also plan to gain further insight into the causes of variability in model performance through a qualitative analysis of outputs.

Acknowledgements

This work is partially supported and funded by Spanish Government project PID2023-151536OB-I00 and by the INCIBE's strategic SPRINT (Seguridad y Privacidad en Sistemas con Inteligencia Artificial) C063/23 project with funds from the EU-NextGenerationEU through the Spanish government's Plan de Recuperación, Transformación y Resiliencia.

References

[1] Shubhi Asthana et al. "Adaptive PII Mitigation Framework for Large Language Models". In: *arXiv preprint arXiv:2501.12465* (2025).

[2] Shubhi Asthana et al. "Deploying Privacy Guardrails for LLMs: A Comparative Analysis of Real-World Applications". In: *arXiv preprint arXiv:2501.12456* (2025).

[3] Yuntao Bai et al. "Constitutional ai: Harmlessness from ai feedback". In: *arXiv preprint arXiv:2212.08073* (2022).

[4] Felix Böhlin. *Detection & Anonymization of Sensitive Information in Text: AI-Driven Solution for Anonymization*. 2024.

[5] Rajeev Chandran and Mei-Ling Tan. "Efficiently Scaling LLMs Challenges and Solutions in Distributed Architectures". In: *Baltic Multidisciplinary Research Letters Journal* 2.1 (2025), pp. 57–66.

[6] Tzuhao Chen, Mila Gascó-Hernandez, and Marc Esteve. "The adoption and implementation of artificial intelligence chatbots in public organizations: Evidence from US state governments". In: *The American Review of Public Administration* 54.3 (2024), pp. 255–270.

[7] Gaia Gambarelli, Aldo Gangemi, and Rocco Tripodi. "Is your model sensitive? SPeDaC: A new benchmark for detecting and classifying sensitive personal data". In: *arXiv preprint arXiv:2208.06216* (2022).

[8] Abel Salinas and Fred Morstatter. "The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance". In: *arXiv preprint arXiv:2401.03729* (2024).

[9] Melanie Sclar et al. "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting". In: *arXiv preprint arXiv:2310.11324* (2023).

[10] Daniel J Solove. *Understanding privacy*. Harvard university press, 2010.

[11] Anh Truong, Austin Walters, and Jeremy Goodsitt. "Sensitive data detection with high-throughput neural network models for financial institutions". In: *arXiv preprint arXiv:2012.09597* (2020).

[12] Aurelian Tutuianu et al. *Efficient statistical techniques for detecting sensitive data*. US Patent 11,599,667. 2023.

[13] Jing Wei et al. "Leveraging Large Language Models to Power Chatbots for Collecting User Self-Reported Data". In: *Proceedings of the ACM on Human-Computer Interaction* 8.CSCW1 (Apr. 2024), pp. 1–35. ISSN: 2573-0142. DOI: 10.1145/3637364. URL: http://dx.doi.org/10.1145/3637364.

[14] Jianliang Yang et al. "Exploring the application of large language models in detecting and protecting personally identifiable information in archival data: A comprehensive study". In: *2023 IEEE International Conference on Big Data (BigData)*. IEEE. 2023, pp. 2116–2123.

[15] Zhiping Zhang et al. ""It's a Fair Game", or Is It? Examining How Users Navigate Disclosure Risks and Benefits When Using LLM-Based Conversational Agents". In: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–26.