Post Hoc Agentic Refinement for Improving Precision in Multilingual Clinical Text De-identification

Justin Xu[1,2,*], **Alistair E W Johnson**[2], **Thomas Lin**[2],
David W Eyre[1], **Rodolfo Quispe**[2]
[1]University of Oxford, [2]Microsoft Health & Life Sciences

Abstract

De-identification systems prioritize recall to protect privacy, but excessive over-tagging reduces data utility. We propose an agentic refiner that reviews high-recall annotations using lightweight tools (validation functions, adaptive context retrieval, persistent to-do state, and modular review skills) to improve precision while minimizing recall loss. Experiments across three multilingual datasets show that the agent achieves significant improvements to binary precision. To support fine-grained analysis, we further introduce a synthetic error dataset of common and systemic failure modes, on which the agent corrects >99% of injected errors in the medical datasets. Our results suggest that agent-based refinement provides a flexible and effective mechanism for improving de-identification precision as a modular extension to existing high-recall systems.

1 Introduction

De-identification of text aims to detect and remove or mask Protected Health Information (PHI) in clinical data and, more broadly, Personally Identifiable Information (PII) in unstructured text. This task is a prerequisite for data sharing, secondary use, and deployment of downstream NLP systems in sensitive domains. From a privacy perspective, recall is paramount: failing to identify a true PHI/PII instance risks irreversible disclosure, whereas over-tagging benign text is less harmful in terms of privacy. Consequently, many de-identification systems are optimized to maximize recall, often at the expense of precision (Uzuner et al., 2007; Stubbs and Uzuner, 2015; Li et al., 2024). However, in many real-world settings, practitioners may be willing to trade small amounts of recall for meaningful improvements in precision, particularly when working within systems with additional safeguards for patient privacy. Excessive over-tagging degrades data utility, harms readability, and can negatively impact applications such as clinical coding, summarization, retrieval, and generative model training. The ideal de-identification system should thus maintain both high recall and high precision – a balance that remains challenging to achieve in scenarios requiring nuanced contextual reasoning (Yang et al., 2019; Li et al., 2024).

In this work, we study post hoc refinement of de-identification outputs as a complement to existing high-recall systems. Rather than introducing a new end-to-end model, we assume an initial set of PHI/PII annotations and focus on improving precision. As a baseline, we consider a simple LLM-as-a-judge approach that reviews annotation spans in context. We then introduce an agentic refiner, designed to reason over annotations using Model Context Protocol (MCP) (Hou et al., 2025) tools and iterative decision-making, enabling more consistent refinement. We additionally construct a synthetic error dataset for fine-grained evaluation of error detection and correction.

Our contributions can be summarized as follows:

1. We propose a modular agent-based refinement pipeline that can be attached to the output of any existing de-identification system to improve precision without retraining (Figure 1).
2. We evaluate this approach across three complementary datasets spanning Spanish clinical notes, German medical transcripts, and English general-domain text with PII (Table 1).
3. We introduce methods to create a synthetic error dataset for de-identification refinement, evaluating error detection (Section 5).

2 Related Works

Early de-identification systems relied on rule-based methods using handcrafted patterns and dictionaries (Neamatullah et al., 2008). Later work framed

[*]Work done during internship at Microsoft

Proceedings of the 25th Workshop on Biomedical Language Processing (BioNLP 2026), pages 115–127

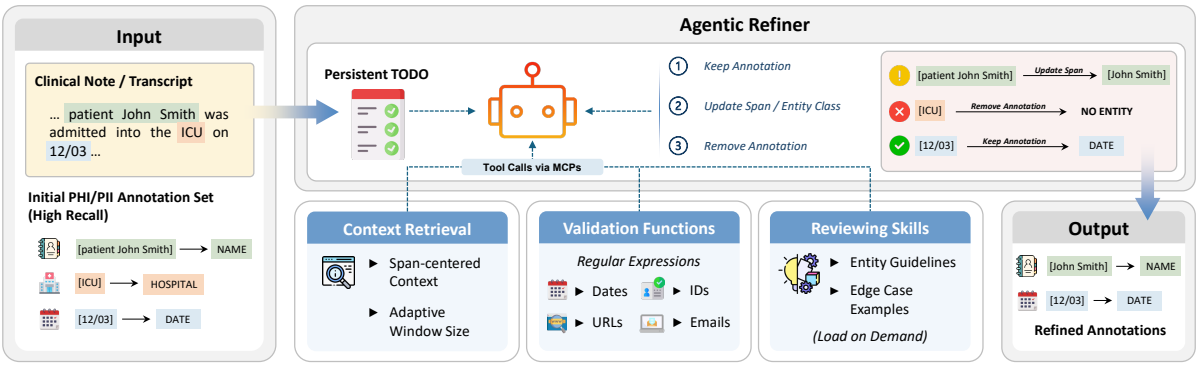

Figure 1: Schematic of Post Hoc Agentic Refinement of Annotations in Clinical Text De-identification.

de-identification as a supervised sequence labeling task, adopting Transformer-based models (BERT and its clinical variants (Alsentzer et al., 2019; Johnson et al., 2020; Huang et al., 2020; Lee et al., 2025; Sounack et al., 2025)). Prior efforts have also combined neural models with external knowledge to generate higher-precision silver data (Tedeschi et al., 2021). More recently, LLMs have been applied to de-identification in zero/few-shot settings, demonstrating acceptable recall and flexibility (Singh et al., 2025; Dai et al., 2025; Zhu et al., 2025; Ding et al., 2025; Aghakasiri et al., 2025). However, LLM-based methods often over-predict PHI/PII and exhibit inconsistent labeling, which is highly sensitive to linguistic nuances in the provided prompt. Agentic systems have also been explored for end-to-end information extraction and clinical reasoning (Yao et al., 2023; Shen et al., 2023; Lu et al., 2025; Lin et al., 2025), but remain costly and difficult to control for de-identification.

In contrast, our work focuses on agent-based refinement rather than replacement. By operating only on an existing set of annotations, our approach provides guardrails to an otherwise open-ended NER task. This modular design allows the agent to be integrated as a precision-enhancing layer on top of high-recall de-identification pipelines.

3 Experimental Design

3.1 Problem Definition

Let a document be represented as a token sequence $D = \{w_1, \ldots, w_n\}$. An upstream de-identification system produces an initial set of PHI/PII annotations $\mathcal{A} = \{a_1, \ldots, a_K\}$, where each annotation $a_i = (s_i, e_i, c_i, t_i)$ consists of a start index s_i, end index e_i, an entity class $c_i \in \mathcal{C}$, and a surface text span $t_i = \{w_{s_i}, \ldots, w_{e_i}\}$.

Given the document D and an associated annotation set $\mathcal{A}$, the refinement task aims to produce an updated set $\mathcal{A}^*$ by evaluating each a_i independently. For a given annotation, the refiner decides whether to (i) retain it unchanged, (ii) update its span boundaries (s_i, e_i), (iii) modify its class c_i, or (iv) remove it entirely if it does not correspond to PHI/PII. The refiner does not introduce new annotations; its sole role is to improve the quality of an existing high-recall annotation set.

This formulation treats de-identification as a contextualized span validation problem rather than end-to-end sequence labeling, enabling refinement to be applied as a post hoc layer on top of arbitrary upstream systems.

3.2 Datasets and Evaluation Metrics

We evaluate annotation refinement across three multilingual datasets. For all datasets, refinement is applied to an initial set of PHI/PII annotations produced by GPT-5 (`gpt-5-2025-08-07`) in a few-shot setting, with entity class-specific guidelines (prompt available in Appendix A.1).

CARMEN-ES (5,614 Total PHI). Derived from the Spanish subset of the *CARMEN-I* corpus of 2,000 documents (Farre Maduell et al., 2024), comprising of discharge summaries, referrals, and radiology reports from Hospital Clínic of Barcelona (March 2020 – March 2022). The collection primarily covers COVID-19 patients with diverse comorbidities. Documents in Catalan or mixed Spanish-Catalan were excluded, and high-quality gold labels were previously validated by experts.

Transcripts-DE (238 Total PHI). Conversational clinical encounters in the German language simulated by medical actors and grounded in real-world patient cases from a hospital in Germany.

WikiPII-EN (24,145 Total PII). Info-boxes with people and animal descriptions derived from English Wikipedia articles (Wang et al., 2018). This represents a general-domain de-identification setting and reflects common non-clinical applications.

Evaluation. Our primary focus is binary PHI/PII detection. An annotation is considered correct if its span overlaps with a gold PHI/PII span, regardless of entity class. We report precision, recall, and F1 where precision measures the proportion of predicted PHI/PII tokens that are correct and recall measures the proportion of gold PHI/PII tokens that are recovered.

3.3 LLM Refiner Baseline

As a baseline, we implement an LLM-based refiner using GPT-5 (`gpt-5-2025-08-07`, medium verbosity and reasoning effort). For each annotation a_i, the model is prompted with the document D, the surface text span t_i from (s_i, e_i), its class c_i, and general PHI/PII labeling guidelines, and is asked to determine whether the annotation should be kept, modified, or removed (prompt available in Appendix A.2). Few-shot examples are provided to encourage consistent behavior.

As seen in "Post-Refinement (GPT-5, Single Call)" of Table 1, while this approach improves binary precision, it removes many valid PHI/PII spans, leading to substantial recall degradation. In addition, the method requires one large API call per annotation with long contextual inputs, motivating a more optimized structured refinement approach.

4 Agentic Refiner

We model annotation refinement as an agentic decision process operating over a document D and its initial annotation set $\mathcal{A} = \{a_1, \ldots, a_K\}$. The agent similarly processes annotations sequentially and produces a refined set $\mathcal{A}^*$.

At each step, the agent maintains a state $\sigma_i = (D, a_i, h_i)$, where $a_i = (s_i, e_i, c_i, t_i)$ is the current annotation under review and h_i denotes the agent's internal memory, including prior decisions and intermediate observations. Given σ_i, the agent selects an action o_i from a discrete refinement space corresponding to retaining, modifying, or removing a_i. Actions may involve tool calls executed through MCP servers, after which the agent updates its state and proceeds to the next annotation.

Formally, the agent implements a policy $\pi(o_i \mid \sigma_i)$ that interleaves natural language reasoning with tool use, producing $\mathcal{A}^*$ after all annotations have been reviewed. Unlike the LLM baseline, the agent's decisions are informed by persistent memory and structured tool outputs, enabling more consistent refinement across long documents. Appendix B highlights representative agentic outputs.

Persistent To-Do List. The agent initializes a persistent to-do list containing all annotations in $\mathcal{A}$ for a given document and marks items as completed as refinement progresses. This externalized memory allows progress tracking over long clinical notes, which may contain several dozen annotations with overlapping surface forms. Similar task decomposition strategies have been shown to improve reliability and long-horizon performance (Yao et al., 2023; Shen et al., 2023).

Pattern-Based Validation Tools. We provide deterministic validation functions based on regular expressions for entity types with well-defined surface forms, such as dates, email addresses, URLs, and identifiers. When applicable, the agent can invoke these tools to validate or reject an annotation without extended reasoning. Offloading high-confidence decisions to symbolic components can reduce hallucinations and improve grounding in hybrid neuro-symbolic and tool-augmented systems (Gao et al., 2023; Schick et al., 2023).

Adaptive Context Retrieval. Rather than always conditioning on the full document D, the agent can dynamically retrieve a window of context around the annotation span, selecting a variable number of tokens to the left and right of t_i based on (s_i, e_i). This flexibility is particularly important for conversational transcripts, where relevant context may span multiple turns of varying length. Adaptive context selection reduces unnecessary token usage while preserving critical discourse information.

Modular Review Skills. Finally, the agent can load specialized review *skills* on demand (Zhang et al., 2025) – modular guideline documents associated with groups of related entity classes. These skills include entity definitions (identical to the guidelines provided to the baseline refiner) and examples of correct and incorrect annotations (including edge cases). The agent decides when to load a skill during its reasoning process, allowing targeted access to entity class-specific knowledge and enables incremental future development.

Dataset	Precision ↑	Recall ↑	F1 ↑
Original GPT-5 Annotations			
CARMEN-ES	90.76	96.75	93.65
Transcripts-DE	56.01	92.02	69.67
WikiPII-EN	83.03	96.85	89.41
Post-Refinement (GPT-5, Single Call)			
CARMEN-ES	94.31 (+3.55)	93.97 (-2.78)	94.14 (+0.49)
Transcripts-DE	69.96 (+13.95)	83.19 (-8.83)	75.97 (+6.30)
WikiPII-EN	90.72 (+7.69)	92.03 (-4.82)	91.37 (+1.96)
Post-Refinement (Agentic)			
CARMEN-ES	97.95 (+7.19)	95.39 (-1.36)	**96.65** (+3.00)
Transcripts-DE	83.68 (+27.67)	87.39 (-4.63)	**85.49** (+15.82)
WikiPII-EN	94.92 (+11.89)	88.79 (-8.06)	**91.75** (+2.34)

Table 1: Binary performance metrics (precision, recall, and F1) before and after refinement across datasets. Values in brackets denote changes relative to pre-refinement (original annotations) performance metrics.

Across all datasets, agentic refinement with GPT-5 (`gpt-5-2025-08-07`, medium verbosity and reasoning effort) yields substantially larger precision gains than the baseline, while better controlling recall degradation (leading to optimal F1). On CARMEN-ES, the agent more than doubles the baseline's precision gain, while incurring a smaller recall drop (over 6× F1 boost). For the German transcripts, agentic refinement also achieves more than 2× the baseline precision gain at a lower cost in recall points, reflecting a favorable precision-recall trade-off in a challenging conversational setting. On WikiPII, the agent again delivers stronger precision gains and overall F1, albeit with a larger recall reduction, indicating that aggressive refinement is more likely to remove borderline entities in general-domain text.

5 Error Dataset

To facilitate a precise and controlled evaluation of annotation refinement, we construct a synthetic error dataset in which the exact location and type of each error is known. This allows us to directly measure how effectively a refiner identifies and corrects annotation errors (Table 2). Starting from reference annotations, we inject realistic errors into medical datasets using three complementary strategies.

Regular Expressions. First, we define a set of regular expression-based rules that inject common and well-known de-identification errors. These rules simulate mistakes frequently observed in both statistical and neural de-identification systems. In total, 10 such rules were specified and applied deterministically; the full list is provided in Appendix C.

Systemic Errors via Clustering. Second, we identify systemic error patterns made by LLM-based de-identification systems. We collect incorrect annotations produced by LLMs along with their surrounding context and encode them using a sentence transformer (Reimers and Gurevych, 2019). The resulting embeddings are projected into a lower-dimensional space using UMAP (McInnes et al., 2020) to reveal clusters corresponding to recurring error types.

Each cluster is then summarized using GPT-5 (`gpt-5-2025-08-07`) to produce an interpretable description of the underlying error pattern. These summaries are used to define targeted error injections. For example, we observe a recurrent pattern where football/soccer clubs are mislabeled as CITY entities despite clear organizational context.

Errors from Model Test Logs. Finally, we mine actual errors from testing logs of an existing BERT-based de-identifier. From these logs, we extract incorrect entity mappings and construct a vocabulary of realistic errors, such as labeling *diabetes* as a HOSPITAL, *pleural* as a LOCATION, or *general* as a PROFESSION outside of a military context. These errors are injected verbatim, preserving their original surface forms and entity classes.

Error Dataset	Caught by Baseline ↑	Caught by Agent ↑
CARMEN-ES	2,714 / 2,807 (96.69%)	**2,779 / 2,807** (99.01%)
Transcripts-DE	248 / 294 (84.26%)	**292 / 294** (99.32%)

Table 2: Using the two medical error datasets, we evaluate both the baseline refiner and the agentic refiner on their abilities to detect and correct injected errors. The agentic refiner identifies and corrects over 99% of injected errors, clearly surpassing the baseline.

6 Discussion & Conclusion

We introduce an agent-based post hoc refinement layer for de-identification that improves precision of existing annotations by combining external memory, validation tools, adaptive context retrieval, and on-demand review skills.

In practice, excessive redaction can meaningfully degrade text quality. We observed frequent upstream errors involving disease eponyms, gene names, and institutional acronyms being labeled as PHI. Although conservative tagging improves privacy protection, indiscriminate removal can alter clinical meaning or remove medically salient

context. Agentic refinement allows more selective correction of such false positives, helping maintain alignment between de-identified text and its original informational content. This is particularly important when de-identified data are used for downstream modeling or large-scale pretraining.

The agentic pipeline was developed incrementally, with components introduced one at a time. In internal testing, each module yielded modest but consistent improvements in stability or performance. Among them, the modular review skills contributed most to precision gains. These skills encode detailed, class-specific guidelines and edge cases aligned with the annotation protocol used to construct our evaluation sets, enabling more consistent reasoning than a single prompt with context. Persistent memory (implemented as a to-do list) improves completeness and prevents skipped spans during multi-step review which can be a pattern common in contemporary agentic systems. Adaptive context retrieval primarily reduces token usage and unnecessary context, improving efficiency rather than directly affecting accuracy. This modular design allows practitioners to selectively enable components depending on deployment constraints or cost considerations. To contextualize these improvements, Appendix D includes a brief efficiency analysis of the proposed framework, reporting runtime, token usage, number of LLM calls, and approximate inference cost per document. Although the agentic pipeline introduces moderate additional computational overhead relative to the baseline LLM refiner, the increase remains bounded and corresponds to measurable gains.

Ultimately, the agent's improvements are not a trivial exchange of recall for precision. While some recall reduction is expected when removing spurious annotations, the agent consistently yields net F1 gains in overall annotation quality and utility, resolving false positives more selectively than baseline refinement. These results suggest that agentic refinement is a practical mechanism for precision control in high-recall de-identification pipelines.

Limitations

Currently, all experiments rely on a single LLM backend (GPT-5) with fixed inference parameters to reduce variability. While the proposed agentic framework is model-agnostic, performance characteristics may differ with other proprietary or open-weight models, particularly those with more limited reasoning capabilities. Preliminary experiments with the open-weight `gpt-oss-120b` model suggest that the framework generalizes beyond proprietary backbones, consistently improving precision, especially on transcript datasets, while exhibiting some recall trade-offs due to lower-recall upstream annotations in this setting.

Additionally, the agent operates strictly as a post hoc refiner and does not introduce new PHI/PII annotations. As a result, it cannot recover entities missed by the upstream de-identification system, and its recall is inherently bounded by the initial annotation set. Furthermore, although we evaluate across multiple languages and data modalities, broader experimentation across additional institutions, document types, and deployment settings would further strengthen generalizability.

The synthetic error dataset is designed to cover a wide range of common and systemic error types, but it does not exhaust the space of possible de-identification failures. While the datasets enable precise and controlled evaluation, they are intended to be extensible and can be expanded as new error patterns are observed in real-world use.

Finally, the agent's capabilities were implemented incrementally. Persistent task tracking, adaptive context windows, validation tools, and review skills were sequentially added as each module yielded observable performance improvements. A formal ablation study would be valuable for quantifying the isolated contributions of each component.

Acknowledgments

We gratefully acknowledge support from Paul Vozila from Microsoft during manuscript writing. JX acknowledges joint funding from Canadian Institutes of Health Research (CIHR) Project ID 202410BCB-535721-77482 (Bioinformatics and Computational Biology), Nuffield Department of Medicine (NDM), and Oxford University Press (OUP). This project was also supported by the National Institute for Health and Care Research (NIHR) Health Protection Research Unit in Healthcare Associated Infections and Antimicrobial Resistance at the University of Oxford in partnership with the UK Health Security Agency (UKHSA) (NIHR207397) and the NIHR Biomedical Research Centre, Oxford. The views expressed are those of the authors and not necessarily those of the NHS, the NIHR, the Department of Health and Social Care, or UKHSA.

References

Kiana Aghakasiri, Noopur Zambare, JoAnn Thai, Carrie Ye, Mayur Mehta, J. Ross Mitchell, and Mohamed Abdalla. 2025. Not what the doctor ordered: Surveying LLM-based de-identification and quantifying clinical information loss. *Preprint*, arxiv:2509.14464 [cs].

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical BERT embeddings. *Preprint*, arxiv:1904.03323 [cs].

Hong-Jie Dai, Tatheer Hussain Mir, Ching-Tai Chen, Chien-Chang Chen, Hao-Ping Yang, Chung-Hong Lee, Yi-Yun Chou, Yu-Chin Teng, Shalini Gupta, Omkar Panchal, Divyabharathy Ramesh Nadar, Wei-Hsiang Liao, Yu-Chuan Lin, Zi-Rui Zhao, Richard Tzong-Han Tsai, Yung-Chun Chang, and Jitendra Jonnagaddala. 2025. Leveraging large language models for the deidentification and temporal normalization of sensitive health information in electronic health records. *npj Digital Medicine*, 8(1):517. Publisher: Nature Publishing Group.

Zhuojun Ding, Wei Wei, and Chenghao Fan. 2025. Selecting and merging: Towards adaptable and scalable named entity recognition with large language models. *Preprint*, arxiv:2506.22813 [cs].

Eulalia Farre Maduell, Salvador Lima-Lopez, Santiago Andres Frid, Artur Conesa, Elisa Asensio, Antonio Lopez-Rueda, Helena Arino, Elena Calvo, Maria Jesús Bertran, Maria Angeles Marcos, Montserrat Nofre Maiz, Laura Tañá Velasco, Antonia Marti, Ricardo Farreres, Xavier Pastor, Xavier Borrat Frigola, and Martin Krallinger. 2024. CARMEN-i: A resource of anonymized electronic health records in spanish and catalan for training and testing NLP tools.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided language models. *Preprint*, arxiv:2211.10435 [cs].

Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model context protocol (MCP): Landscape, security threats, and future research directions. *Preprint*, arxiv:2503.23278 [cs].

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *Preprint*, arxiv:1904.05342 [cs].

Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, pages 214–221. Association for Computing Machinery.

Simon A. Lee, Anthony Wu, and Jeffrey N. Chiang. 2025. Clinical ModernBERT: An efficient and long context encoder for biomedical text. *Preprint*, arxiv:2504.03964 [cs].

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024. Improving in-context learning of multilingual generative language models with cross-lingual alignment. *Preprint*, arxiv:2311.08089 [cs].

Xinkui Lin, Yuhui Zhang, Yongxiu Xu, Kun Huang, Hongzhang Mu, Yubin Wang, Gaopeng Gou, Li Qian, Li Peng, Wei Liu, Jian Luan, and Hongbo Xu. 2025. MAKAR: a multi-agent framework based knowledge-augmented reasoning for grounded multimodal named entity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6121–6141. Association for Computational Linguistics.

Meng Lu, Yuzhang Xie, Zhenyu Bi, Shuxiang Cao, and Xuan Wang. 2025. CROSSAGENTIE: Cross-type and cross-task multi-agent LLM collaboration for zero-shot information extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13953–13977. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arxiv:1802.03426 [stat].

Ishna Neamatullah, Margaret M. Douglass, Li-wei H. Lehman, Andrew Reisner, Mauricio Villarroel, William J. Long, Peter Szolovits, George B. Moody, Roger G. Mark, and Gari D. Clifford. 2008. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8:32.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *Preprint*, arxiv:1908.10084 [cs].

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Preprint*, arxiv:2302.04761 [cs].

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face. *Preprint*, arxiv:2303.17580 [cs].

Praphul Singh, Charlotte Dzialo, Jangwon Kim, Sumana Srivatsa, Irfan Bulu, Sri Gadde, and Krishnaram Kenthapadi. 2025. RedactOR: An LLM-powered framework for automatic clinical data de-identification. *Preprint*, arxiv:2505.18380 [cs].

Thomas Sounack, Joshua Davis, Brigitte Durieux, Antoine Chaffin, Tom J. Pollard, Eric Lehman, Alistair E. W. Johnson, Matthew McDermott, Tristan Naumann, and Charlotta Lindvall. 2025. BioClinical ModernBERT: A state-of-the-art long-context encoder for biomedical and clinical NLP. *Preprint*, arxiv:2506.10896 [cs].

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58 Suppl:S20–S29.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533. Association for Computational Linguistics.

Ozlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the state-of-the-art in automatic de-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, Tilburg University, The Netherlands. Association for Computational Linguistics.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4431–4441. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. *Preprint*, arxiv:2210.03629 [cs].

Barry Zhang, Keith Lazuka, and Mahesh Murag. 2025. Equipping agents for the real world with agent skills. Published October 16, 2025. Updated December 18, 2025 to announce Agent Skills as an open standard.

Guangcheng Zhu, Ruixuan Xiao, Haobo Wang, Zhen Zhu, Gengyu Lyu, and Junbo Zhao. 2025. Large margin representation learning for robust cross-lingual named entity recognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4270–4291. Association for Computational Linguistics.

A Prompts

A.1 Original Annotation Prompt

The following prompt was used to generate the initial PHI/PII annotations on which all refinement experiments were conducted:

GPT-5

```
You will be provided a note containing named entities.
Please identify all instances of the named entity: {ENTITY}.

Specific annotation instructions for the entity: {ENTITY_INSTRUCTION}

Respond with the entity type and the text of the entity, with one entity per line.
Here is an example for {LANGUAGE}: {EXAMPLE}

Respond with the text from the note EXACTLY.

...

If no annotations are found, respond with NO ENTITIES.

The user will provide the note to annotate.
```

A.2 Baseline Refinement Prompt

The following prompt was used to perform baseline LLM-based refinement of PHI/PII annotations:

GPT-5

```
You are correcting named entity recognition annotations.

You will be provided with a single named entity annotation and the context.

You must respond with one of the following:
    * correct - The provided annotation is correct.
    * update - There is an entity, but it must be corrected in some way.
    * remove - The annotated text does not correspond to the given entity type.

{ENTITY GUIDELINES}

{EXAMPLES}

The user will provide the annotation to correct.
```

B Example Agentic Output

The following examples illustrate representative cases where the single-pass LLM refiner failed to appropriately remove incorrect PHI/PII annotations, either due to insufficient contextual reasoning or inconsistent application of labeling guidelines. In contrast, the agentic refiner successfully identifies and corrects these errors by combining persistent task tracking, targeted context retrieval, deterministic validation, and on-demand review skills. These examples highlight how agentic refinement enables more reliable and selective correction of false positives than prompting alone.

Example 1 (Mentions of Medication Manufacturers). This example illustrates systematic over-tagging of pharmaceutical manufacturers and brand names as ORGANIZATION entities in a therapeutic plan. Although the spans correspond to real companies, the surrounding context makes clear that they function purely as medication descriptors and do not identify patients, providers, or institutions. By loading the organization-specific review skill and retrieving targeted context for each span, the agent consistently recognizes these mentions as non-PHI and removes all annotations. This case highlights the agent's ability to apply contextual guidelines at scale and eliminate large batches of false positives that frequently occur in medication lists.

Original annotations (CARMEN-I PLAN TERAPEUTICO):

Note ID	Index ($s_i - e_i$)	Span	Entity Class
<NOTE_ID>,	29-35,	normon,	ORGANIZATION
<NOTE_ID>,	128-139,	kern pharma,	ORGANIZATION
<NOTE_ID>,	223-228,	davur,	ORGANIZATION
<NOTE_ID>,	333-338,	mylan,	ORGANIZATION
<NOTE_ID>,	685-696,	kern pharma,	ORGANIZATION
<NOTE_ID>,	782-790,	teofarma,	ORGANIZATION
<NOTE_ID>,	912-916,	teva,	ORGANIZATION
<NOTE_ID>,	1181-1186,	cinfa,	ORGANIZATION
<NOTE_ID>,	1279-1284,	cinfa,	ORGANIZATION
<NOTE_ID>,	1380-1387,	tevagen,	ORGANIZATION
<NOTE_ID>,	1606-1611,	pensa,	ORGANIZATION
<NOTE_ID>,	1718-1724,	apotex,	ORGANIZATION

Agentic Refiner

Agent Summary:

- Annotations Reviewed: 12
- Annotations Modified: 0
- Annotations Deleted: 12

Actions Taken:

- Loaded the ORGANIZATIONS skill. Per guidelines, vaccine/pill/device brands and pharmaceutical manufacturer names (e.g., Normon, Kern Pharma, Mylan, Teva, Cinfa, etc.) should NOT be tagged as ORGANIZATION in medication lists, as they are not PHI in this context.
- Retrieved context for each item and confirmed all were medication manufacturer/brand mentions in a therapeutic plan.
- Deleted all 12 ORGANIZATION annotations accordingly.
- Final validation returned no errors. The final annotation list is empty.

Major Issues Resolved: Removal of 12 non-PHI/incorrectly labeled items (organizations that are not PHI in context).

Example 2 (Mixed Clinical Backgrounds). This example demonstrates selective refinement in a document containing a diverse set of entity types, including dates, locations, organizations, and clinical terminology. The agent removes only two incorrectly labeled spans: an ORGANIZATION tag applied to a disease classification token (*GOLD*) and a LOCATION tag applied to an anatomical term (*cisura silviana*). All other annotations are retained after context-aware verification, including correctly labeled hospitals, clinicians, dates, and ages. This example shows that the agent does not indiscriminately trade recall for precision, but instead performs targeted corrections while preserving valid PHI annotations.

Original annotations (CARMEN-I IA ANTECEDENTES):

Note ID	Index ($s_i - e_i$)	Span	Entity Class
...			
<NOTE_ID>,	2453-2457,	GOLD,	ORGANIZATION
...			
<NOTE_ID>,	2918-2933,	cisura silviana,	LOCATION

Agentic Refiner

Agent Summary:

- Annotations Reviewed: 28
- Annotations Modified: 0
- Annotations Deleted: 2

Actions Taken:

- Loaded relevant skills: REVIEW, DATES-AGES, LOCATIONS, NAMES-PEOPLE, PROFESSIONS, ORGANIZATIONS.
- Checked rules for DATE entries; none qualified for automatic acceptance; proceeded with manual review.
- Retrieved surrounding context for all annotations.
- Manually reviewed problematic spans:
  - Deleted ORGANIZATION 'GOLD' - context shows this as a COPD severity/classification term (non-PHI).
  - Deleted LOCATION 'cisura silviana' - identified as an anatomical/fissure term (non-PHI).
- Verified remaining annotations:
  - Ages and professions correctly labeled and contextually PHI.
  - Hospital `C. La Salud` and doctor `Dr. Ager` correctly labeled and complete.
  - Dates (years, month-year, numeric formats, weekdays, and schedule abbreviations `Ma/Ju/Sa`) are tied to patient care timeline; boundaries and labels confirmed.
- Final validation succeeded with no errors or warnings. Remaining annotations marked as correct per guidelines.

Major Issues Resolved: Removal of 2 non-PHI/incorrectly labeled items (classification token and anatomical label).

C Regular Expression Rules for Error Injection

We inject a set of deterministic, regular expression-driven error patterns to simulate common annotation mistakes. Each rule targets a realistic failure mode observed in deployed de-identification systems or LLM annotation outputs. The 10 rules used in our experiments are described below.

Ambiguous common-word names. Many surnames or given names are also common English words (*e.g.*, `Brown`, `May`, `Grant`, or `Jordan`). This rule treats such tokens as NAME/PATIENT entities in contexts where they are not PHI (*e.g.*, "brown" in "the patient reported a brown rash" → incorrectly labeled as PATIENT).

Example

"Brown" ⇒ injected as NAME.
"May" in "the patient may require surgery" ⇒ injected as NAME.
"Grant" in "we will grant discharge tomorrow" ⇒ injected as PATIENT.
"Mann" in "der mann klagt über Schmerzen" ⇒ injected as PATIENT.

Two-digit lab values mislabeled as AGE. Numeric laboratory or microbiology results (often two digits) appear near medical keywords and can be mistaken for patient ages. We identify two-digit numbers near medical tokens and relabel them as AGE.

Example

"12" in "WBC 12" near "culture" ⇒ injected as AGE.
"45" in "platelets 45" near "CBC" ⇒ injected as AGE.
"22" in "RR 22" ⇒ injected as AGE.
"45" in "Leukozyten 45" nahe "Infektion" ⇒ injected as AGE.

Blood-pressure values labeled as DATE. Blood-pressure strings such as "120/80" resemble date-like patterns and may be mis-tagged as DATE. We search for paired numeric tokens with a slash in clinical contexts and relabel them as DATE.

Example

"120/80" in "BP 120/80" ⇒ injected as DATE.
"130/85" in "BP 130/85 sitting" ⇒ injected as DATE.
"90/60" in "hypotensive at 90/60" ⇒ injected as DATE.
"130/85" in "Blutdruck 130/85 gemessen" ⇒ injected as DATE.

City → State/Organization confusion. Place-name granularity errors: city names are relabeled as either broader administrative units or organizations. This simulates errors where classifiers confuse geographic granularity or entity class.

Example

"Berlin" ⇒ injected as STATE.
"Madrid" in "ingresado en Madrid ayer" ⇒ injected as ORGANIZATION.
"Paris" in "treated in Paris clinic" ⇒ injected as ORGANIZATION.

Generic healthcare words → HOSPITAL. Terms that denote wards, units, or specialties (*e.g.*, `ICU`, `cardiology`, or `ward`) are relabeled as full HOSPITAL entities. This models over-generalization errors.

Example

"ICU" ⇒ injected as HOSPITAL.
"cardiology" in "admitted to cardiology" ⇒ injected as HOSPITAL.
"ER" in "sent to ER for evaluation" ⇒ injected as HOSPITAL.

Patient/Doctor confusion. Swap PATIENT and DOCTOR labels to simulate role-disambiguation mistakes (*e.g.*, doctor names annotated as patients and vice versa).

Example

"Dr. Smith" ⇒ injected as PATIENT.
"John Davis" in "Dr. John Davis reviewed labs" ⇒ injected as PATIENT.

Age/Date confusion. Swap AGE and DATE labels to mimic numeric ambiguity (*e.g.*, two- or four-digit tokens interpreted as a birth year vs. age).

Example

"1978" in "Born 1978" ⇒ injected as AGE.
"84" in "patient is 84" injected as DATE.
"63" in "63-jähriger Patient" ⇒ injected as DATE.

Hospital/Organization confusion. Map specific hospital names to a generic ORGANIZATION label intended for employers/companies, simulating coarse-grained classification errors.

Example

"Hospital Clínic" ⇒ injected as ORGANIZATION.
"Massachusetts General Hospital" ⇒ injected as ORGANIZATION.
"Universitätsklinikum Heidelberg" ⇒ injected as ORGANIZATION.

ZIP / City / State swaps. Exchange geographic and postal labels (ZIP, city, or state) to reflect location-granularity errors and numeric/non-numeric confusions.

Example

"02115" (ZIP) ⇒ injected as CITY.
"CA" ⇒ injected as CITY.

Eponymous diseases as PATIENT. Diseases named after people (*e.g.*, `Parkinson's` or `Alzheimer's`) are injected as PATIENT entities, modeling a frequent confusion where models treat eponyms as person names.

Example

"Parkinson" in "Parkinson's disease" ⇒ injected as PATIENT.
"Hodgkin" in "Hodgkin lymphoma" ⇒ injected as PATIENT.
"Morbus Alzheimer" ⇒ injected as PATIENT.

Procedures. Each rule is applied deterministically over reference annotations or raw text according to a priority order to avoid conflicting injections (*e.g.*, numeric-format rules are applied before generic city/state swaps). Injected errors are recorded with provenance (rule id, original label, injected label) to enable precise evaluation when using the error dataset.

These rules are designed to be realistic but conservative: they target representative failure modes rather than exhaustively enumerating every possible error. The rule set is extensible; new patterns discovered in deployment logs can be added to the vocabulary and re-run to produce larger or more diverse synthetic error sets.

D Comparison of Computational Costs

Metric	LLM Baseline	Agentic Refinement
# LLM Calls	420	161
Time to Process (seconds)	1,124	1,820
Input Tokens	2,284,378	3,076,333
Output Tokens	432,270	720,450
Approx. Cost (USD)	$7.18	$11.05

Table 3: Efficiency comparison between the single-pass LLM refinement baseline and the proposed agentic refinement pipeline on a subset of the CARMEN-ES dataset (3,314 tokens; 84 annotations). Wallclock time reflects end-to-end single-threaded execution, including preprocessing and postprocessing. Costs are approximate and based on standard GPT-5 pricing.