SPY: Enhancing Privacy with Synthetic PII Detection Dataset

Anonymous COLING 2025 submission

Abstract

We introduce SPY dataset: a novel synthetic dataset for the task of Personal Identifiable Information (PII) detection, underscoring the significance of protecting PII in modern data processing. Our research innovates by leveraging Large Language Models (LLMs) to generate a dataset that emulates real-world PII scenarios. Through evaluation, we validate the dataset's quality, providing a benchmark for PII detection. Comparative analyses reveal that while PII and Named Entity Recognition (NER) share similarities, dedicated NER models exhibit limitations when applied to PII-specific contexts. This work contributes to the field by making the generation methodology and the generated dataset publicly accessible, thereby enabling further research and development in this field.[1]

1 Introduction

In the expanding digital realm, the accumulation of personal data has reached unprecedented levels. Details encompassing our search queries, online activity, social connections, health records, and more are gathered and disseminated among advertisers, researchers, and governmental bodies, giving rise to complex privacy concerns about keeping personal information safe. What entities qualify as personally identifiable information? For example, a Social Security number (SSN) is undoubtedly considered PII, but is a person's name considered PII? Narayanan and Shmatikov (2010) argues that PII is surprisingly difficult to define.

Historically, NER techniques have been employed for PII detection. However, when security is a primary concern – and it usually is in the context of PII detection – PII entities constitute a subset of NER entities. For instance, a person's name on a credit card is clearly PII, and revealing this information can indeed cause harm. Conversely, the name of the lead actress in the Titanic movie, would likely not cause any harm upon disclosure. In this work, we define PII entities as those that can be used to identify, contact, or locate a specific individual and should not be disclosed to the public due to security concerns. The distinction between PII and NER entities is outlined in Table 1.

	PII vs NER
a)	Apple technical support for education customers: 1-800-800-2775. Satya Nadella is CEO of Microsoft Corp.
b)	Mark Lee lives at 80081 Hardy Crescent Suite 420.

Table 1: Examples of NER entities - section **a)** and PII entities - section **b)**.

If PII detection and NER are distinct, it implies that data-driven approaches for PII detection require their own specialized dataset. However, creating and sharing a dataset with actual PII entities online is not feasible due to privacy concerns. Consequently, there are two options: (1) utilize a dataset containing real PII entities and substitute them with fake ones generated by Faker (Faraglia, 2014); (2) devise a methodology to entirely generate a PII-focused dataset from scratch. The benefit of the former approach is that it maintains the data's characteristics. The drawback is ensuring that all genuine PII entities have been accurately replaced.

In our work, we opted for the second approach. We employed Faker to create artificial PII entities and Llama-3-70B to generate text where these fake entities could be seamlessly integrated.

The additional advantage of the fully generated approach lies in having complete control over the generation process. You can tailor it to your specific domain, including designated PII entities and their desired distribution or balance.

[1] We will release the code and the dataset to facilitate future research of PII.

Our contributions can be summarized as follows:

- We present a methodology for developing a SPY dataset and compare it to other methodologies used for creating a synthetic PII datasets. Our approach does not require any external data and can be applied to any knowledge domain.

- We open-source a SPY dataset containing 4,491 medical consultations and 4,197 questions in legal domain, that is specifically developed to highlight the contrast between an average task of named entity recognition and more fine-grained tasks of PII detection.

2 Related Work

Traditional methods for safeguarding PII include encryption, anonymization, and pseudonymization, which help to mask or replace identifiable information. Knowledge-based approaches like `regexp` achieve fair accuracy in identifying phone numbers, emails (Aura et al., 2006), IP addresses (Allal et al., 2023) and other PII types that have a strict and template-based format. However, while these techniques perform adequately for structured data, they fall short when applied to unstructured text. This is where data-driven approaches, like Named Entity Recognition (NER), come into play. NER models offer greater flexibility in identifying PII across varied contexts, particularly when dealing with unstructured data like names or addresses, by learning from labeled datasets containing examples of PII instances (Johnson et al., 2020; Pilán et al., 2022; Li et al., 2023).

Detecting PII requires identifying entities that pose potential privacy risks, which may not always align with conventional NER categories. While rule-based methods remain useful for detecting PII with rigid formats, the ambiguity between personal and non-personal entities coupled with the complexities of natural text calls for machine learning techniques.

Existing PII detection tools and datasets often fail to distinguish between personal and non-personal entities within the same entity type, essentially performing as traditional NER systems. For example, Microsoft's **Presidio** (Microsoft, 2021), a popular tool for PII detection, combines NER models with regular expressions and pattern matching. However, this approach labels all entities of a given type (e.g., names) as PII, without differentiating between personal and non-personal entities. Similarly, **NERPII** (Mazzarino et al., 2023), a pseudonymization tool for structured data, leverages Presidio and BERT (Devlin et al., 2019) for PII detection but shares the same limitations.

One of the major challenges in PII detection is the scarcity of publicly available datasets due to privacy concerns. To address this, some approaches replace personal data in real texts with synthetic data, while others generate entirely synthetic texts. Below are some of the more popular datasets for PII detection:

The **BigCode**[2] PII dataset was created from The Stack (Kocetkov et al., 2022) dataset, containing code from open-source GitHub repositories. It focuses on identifying PII in code, such as keys, IP addresses, and emails, using tools like detect-secrets for pre-filtering. The dataset was annotated by 1,399 crowd-workers via the Toloka platform. Although useful for PII detection in code, BigCode is limited to programming environments, making it less suitable for broader text-based PII scenarios. Additionally, manual annotation may hinder scalability.

The **AI4Privacy**[3] dataset is a synthetic PII dataset created using proprietary algorithms, ensuring no privacy violations. It spans six languages and eight jurisdictions, with 63 PII classes, making it one of the most comprehensive datasets available. However, despite its wide applicability, the underlying data generation algorithms are closed-source, limiting transparency in how the dataset was constructed. This lack of transparency makes it difficult to assess the representativeness of the data or adapt it to specific needs.

The **Kaggle PII Detection Competition** (Langdon et al., 2024) dataset, provided by The Learning Agency Lab, contains around 22,000 student essays from a massive open online course. Unlike other PII datasets mentioned earlier, this one distinguishes between PII and non-PII entities, aligning more closely with the goal of this research. However, it has two significant limitations. First, all the essays are written in response to a single assignment prompt, which limits the diversity of the data and, consequently, the variety of PII scenarios. Second, only 30% of the dataset is publicly available for training, with the remaining 70% reserved for testing, making it unsuitable for a comprehen-

[2] hf.co/datasets/bigcode/bigcode-pii-dataset
[3] hf.co/datasets/ai4privacy/pii-masking-300k

sive evaluation (see Figure 2 for detailed statistics). While the dataset provides accurate PII annotations for seven entity types, these limitations in diversity and access make it less ideal for broader applications and thorough evaluations.

2.1 Synthetic NER Generation

Although research on PII datasets is limited due to privacy concerns, there has been significant work on generating synthetic NER datasets, which share a similar format with PII data. Therefore, we provide an overview of techniques used for synthetic NER data generation, as they are relevant to PII detection.

One notable approach is outlined by Tang et al. (2023), where a small set of human-labeled examples is used to guide LLMs in generating diverse synthetic datasets. This method encourages variability in sentence structures and linguistic patterns, ensuring that the synthetic data is not overly repetitive or predictable. A post-processing step is then employed to filter out low-quality or duplicate samples, ultimately improving the quality and diversity of the data.

Another promising technique involves automatic data annotation, where synthetic data is used to enrich an existing labeled dataset. Tools like **UniNER** (Zhou et al., 2024) and **NuNER** (Bogdanov et al., 2024a) leverage GPT-3.5 to annotate large text corpora, such as The Pile (Gao et al., 2020) and C4 (Raffel et al., 2020). These models are pretrained on these annotated corpora to create versatile, general-purpose NER models, which can then be fine-tuned with a smaller amount of domain-specific data.

3 Data Construction

While directly prompting LLMs to annotate text data has proven effective for datasets rich in NER entities (Zhou et al., 2024; Bogdanov et al., 2024b; Zaratiana et al., 2023), this approach is less effective in data-scarce environments. When only a small fraction of the dataset contains PII entities, LLM-based annotation becomes less efficient due to several challenges: (1) only a small portion of texts in dataset will receive any annotations, (2) certain entity types will be underrepresented, and (3) the resulting annotations will be highly imbalanced across classes. For example, in the Kaggle competition dataset mentioned earlier, only 24% of student essays contain personal data, and

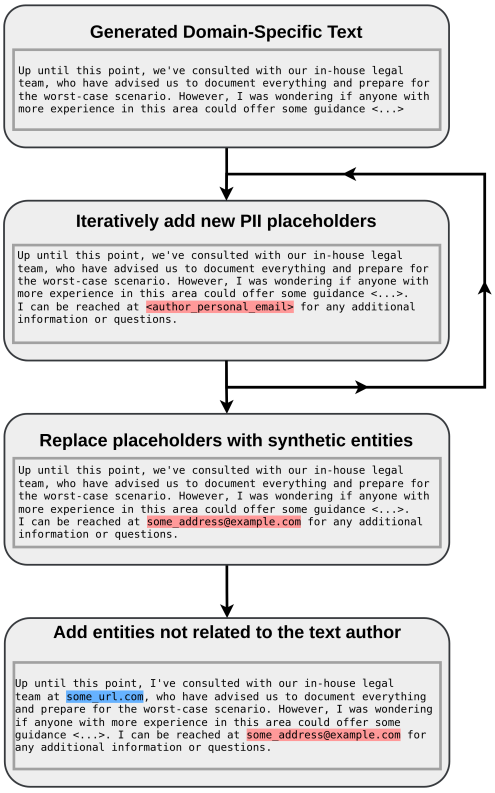

Figure 1: Multi-step prompting procedure. Red selection – author's personal data (PII); blue selection – NER entities not directly related to the text author.

six out of the seven entity types have fewer than 110 annotations (see Figure 2), leading to class imbalance and limited representation. To address those constraints, we generate texts already containing predetermined sets of personal entities. We chose two domains: (1) legal – informal questions in legal domain similar to r/LegalAdvice[4] and (2) medical – forms completed by patients for online medical consultations. Specifically, we select the following PII entity types: `name`, `email`, `phone number`, `personal url`, `personal identifier`, `username`, and `personal address`.

3.1 Prompting Pipeline

When designing a methodology for generating texts with personal data, it's important to clearly distinguish PII entities from other types of information. Any details about the text's author can be classified as PII, while information that can be referenced through links to web resources, papers, or articles is considered publicly available. Based on this distinction, we chose to prompt the LLM to generate only PII placeholders related to the text author. In

[4] https://www.reddit.com/r/legaladvice/

	Legal questions			Medical Consultations		
	upd pii 1	upd pii 2	final	upd pii 1	upd pii 2	final
name	0.58	1.06 (+0.48)	0.91 (+0.33)	0.69	1.12 (+0.43)	0.99 (+0.3)
email	1.03	1.15 (+0.12)	0.86 (-0.17)	1.01	1.12 (+0.11)	0.93 (-0.08)
username	0.91	1.14 (+0.23)	1.30 (+0.39)	0.80	1.16 (+0.36)	1.33 (+0.53)
phone number	0.87	1.1 (+0.23)	0.75 (-0.12)	0.88	1.12 (+0.24)	0.89 (+0.01)
personal url	1.07	1.34 (+0.27)	0.87 (-0.2)	1.03	1.32 (+0.29)	0.88 (-0.15)
personal address	0.71	1.19 (+0.48)	0.87 (+0.16)	0.73	1.28 (+0.55)	1.06 (+0.33)
personal identifier	0.39	0.98 (+0.59)	0.69 (+0.3)	0.53	1.05 (+0.52)	0.89 (+0.36)
average	0.79	1.14 (+0.35)	0.89 (+0.1)	0.81	1.17 (+0.36)	0.99 (+0.18)

Table 2: Average number of PII entities detected in texts generated by SPY prompting pipeline. Each entity type is counted separately. *upd pii {k}* refers to the average number of PII entities in texts after k iterative updates of PII placeholders using the prompt from Figure 7; *final* represents the average number of PII entities in texts that have completed all stages of the pipeline.

contrast, all non-PII entities are unrelated to the author. These constraints helped ensure a clear separation between personal and publicly available information.

SPY prompting methodology was developed to meet the following criteria: (1) incorporate domain-specific details while naturally integrating PII entities, (2) include both personal and non-personal entities from predefined categories, and (3) maintain a clear distinction between personal and public data. To achieve this, we implemented a multi-stage prompting pipeline, as shown in Figure 1.

First, we used the Llama-3-70B model to generate texts in the *law* and *medical* domains, following the prompt in Figure 6. To enhance the diversity of the generated texts, we included details about the person's occupation and personality, which expanded the range of topics within each domain. Personalities were generated using the prompt shown in Figure 5.

When incorporating the author's personal information, we encountered difficulties embedding multiple PII entities at once. To address this, we adopted an iterative approach, prompting the model to refine each version of the text, progressively adding more entities as outlined in Figure 7. While iterative text updates can be done using CoD prompting (Adams et al., 2023), we found that Llama-3-70B struggled with applying multiple updates in a single generation due to the length of the initial texts. Additionally, instead of directly inserting PII, we used placeholders (`<entity-type>`) during generation to minimize paraphrasing.

Before proceeding to the next stage, we replaced all placeholders with corresponding synthetic entities to ensure consistency between previously added PII and new entities. The `Faker` library (Faraglia, 2014) was used to generate a diverse set of synthetic personal entities, localized to six different countries.

After completing this process, we obtained a dataset with personal information exclusively tied to the text author. In the final stage, we introduced non-PII entities unrelated to the author using the prompt in Figure 8.

4 Data Analysis

SPY's flexible pipeline for synthetic PII data generation demonstrates several key advantages:

- **Even Distribution of PII Entities**: The pipeline ensures that PII entities are evenly distributed throughout the generated texts. This even distribution is visually represented in the Figure 3 where the entities' positions are spread relatively uniformly across the texts, avoiding clustering in any specific section.

- **Balanced Entity Counts**: The number of entities by type is relatively balanced. For example, we observed that after running the pipeline, there were approximately 3,000–4,000 entities for every entity type, showing that the dataset maintains a fair balance across different types of PII entities. For more detailed statistics see Figure 3.

- **Controlling PII Entity Density**: The iterative update mechanism allows us to increase the number of PII entities in generated texts by repeating the update step multiple times. In

Table 2 in column *upd pii 2* we can see a stable increase in the average number of entities present in text. This flexibility in entity injection enables the generation of more entity-rich texts. We opted against more than 2 updates to avoid compromising the natural flow of the text through excessive inclusion of personal information.

- **Controlling non-PII Entities**: Another significant benefit of this pipeline is the ability to control the inclusion of non-PII entities, such as public names, organizations, or general locations. This degree of control would not be possible if real text data were simply marked up using a tool like ChatGPT, as that approach would not allow for the same precision in distinguishing between personal and non-personal data. However a major limitation is that while generating non-PII entities LLaMA-3-70B tends to drop some of the previousely generated PII placeholders as shown in Table 2 in column *final*.

Entity type	Domain	
	Legal questions	Medical Consultations
url	3281	3388
email	3707	4065
username	3854	3968
address	3592	4121
name	4046	4276
phone number	3304	3847
id_num	3186	4024

Table 3: Number of generated PII entities by type.

The pipeline thus provides a robust solution for generating synthetic data with controlled distributions, balancing the number of entities while ensuring flexibility in both PII and non-PII management.

5 Experiments

5.1 Baselines

Below, we outline several zero-shot baseline approaches we employ for PII detection.

Presidio (Microsoft, 2021) is Microsoft SDK that provides a fast identification for PII entities such as credit card numbers, names, locations, phone numbers, financial data and more. Presidio employs a combination of techniques to detect PII including leverages NER module, regular expressions, rule based logic and checksum with relevant context in multiple languages.

LLaMA-3-70B (AI@Meta, 2024) is a LLM that we employ with zero-shot instructions for extracting personal entities. LLaMA-3-70B, like other LLMs, is highly effective at following instructions and well-suited for NER tasks. This model processes and identifies a wide range of personal information directly from text, demonstrating strong adaptability and generalization across different types of personal entities.

5.2 Experimental setup

Our solution is based on DeBERTa (He et al., 2021) model, which is recognized as state-of-the-art for token classification tasks. It outperforms BERT and RoBERTa models by introducing a disentangled attention for encoding the position and content of each token separately into two vectors. In our experiments, we fine-tuned `DeBERTaV3-base` (He et al., 2023) , which is an enhanced version of DeBERTa. The hyperparameters for the model can be found in Appendix A.

5.3 Evaluation

For our evaluation, we use micro-averaged Precision, Recall, and F1-score, which are standard metrics for assessing token classification tasks. We provide Precision-Recall curves to offer a more comprehensive view of our model's performance.

Not having a manually annotated test set, we calculate metrics for our DeBERTa model in a domain-transfer setting by training it on one domain and calculating test scores on another.

5.4 Experimental results

First, we verify that SPY contains a substantial amount of non-PII entities. To do this, we evaluated the UniNER (Zhou et al., 2024) model on the *NAME* entity type using the prompt shown in Table 5. The results indicate that Recall is significantly higher than Precision, suggesting that UniNER identified additional non-PII names. This observation is further supported by the example provided in Table 7.

Following the presented pipeline, we generated two datasets from legal and medical domains, each containing 4,000+ texts. Table 4 shows how different models perform PII detection on a SPY dataset.

Presidio is not capable of recognizing usernames and addresses limiting their effectiveness in PII detection tasks. We can clearly see that Presidio has a much lower Precision than the Recall for all the categories, meaning that it misclassified a large

Entity	Metric	Medical Consultations			Legal Questions		
		Llama-3-70B	Presidio	DeBERTa (our)	Llama-3-70B	Presidio	DeBERTa (our)
Name	P	73.0	17.1	**86.9**	64.7	17.9	**87.4**
	R	62.9	80.4	**88.7**	68.9	79.4	**93.2**
	F1	67.6	28.2	**87.8**	66.7	29.2	**90.2**
Email	P	92.7	37.6	**97.6**	91.8	33.7	**92.1**
	R	90.9	92.2	**99.5**	88.5	91.8	**99.1**
	F1	91.8	53.4	**98.5**	90.1	49.3	**95.5**
Username	P	68.8	-	**92.1**	66.1	-	**90.3**
	R	70.4	-	**95.4**	59.7	-	**98.0**
	F1	69.6	-	**93.8**	62.7	-	**94.0**
URL	P	83.6	6.9	**97.5**	84.5	7.9	**94.4**
	R	91.9	19.4	**98.9**	92.5	21.3	**99.0**
	F1	87.5	10.2	**98.2**	88.3	11.5	**96.7**
ID	P	91.7	26.1	**96.7**	91.9	20.6	**93.0**
	R	75.1	38.9	**98.3**	62.2	34.4	**96.6**
	F1	82.6	31.2	**97.5**	74.2	25.8	**94.8**
Phone	P	89.8	37.4	**93.3**	85.7	34.1	**87.5**
	R	90.0	65.5	**96.9**	92.8	68.1	**98.7**
	F1	89.9	47.6	**95.0**	89.1	45.4	**92.8**
Address	P	**96.2**	-	89.3	**93.7**	-	88.3
	R	90.4	-	**95.1**	81.3	-	**94.5**
	F1	**93.2**	-	92.1	87.1	-	**91.3**

Table 4: Performance metrics of models with various domain and entities, where P – Precision, R – recall, F1 – F-score. **Presidio** is a Microsoft SDK for fast PII detection using NER, regex, rule-based logic. **LLaMA-3-70B-zero-shot** is a zero-shot prompted LLM for PII task. **DeBERTa** is a model cross-validated on different domains of SPY dataset. Blanks mean that entity class is not supported by the model.

Legal questions			Medical Consultations		
P	R	F1	P	R	F1
21.5	89.5	34.7	21.7	80.4	34.1

Table 5: UniNER evaluation results on the SPY dataset. Metrics are calculated specifically for entities of the *NAME* type, using prompts from the original UniNER paper (Zhou et al., 2024): *"What describes a person in the text?"*

portion of NER entities as PII entities. Another observation is that Llama-3-70B consistently outperforms Presidio, which can be attributed to its ability to differentiate between standard NER entities and PII entities.

DeBERTa validated on a SPY dataset in a domain-transfer setting is able to detect PII entities more precisely than zero-shot methods, getting a much higher Precision with a smaller gap between Recall. Encoder model specifically trained to detect PII entities outperforms general NER models confirming the fact that the task of PII detection is not equivalent to NER. The distinction between them can be effectively learned by a supervised classification model.

6 Conclusions and Discussion

In this study, we delve into the critical issue of PII detection, highlighting its importance in the realm of data privacy and security. We underscore the distinction between PII detection and NER, emphasizing that while related, PII detection carries unique nuances and requirements.

We highlighted the disadvantages of existing datasets and PII tools and provide a robust methodology for creating diverse training datasets tailored for PII detection. Our approach is based on employing LLM to generate data and doesn't require human supervision. These advancements reinforce our commitment to safeguarding personal data, a significant area in today's digital landscape. To encourage the research in the field we make the SPY dataset freely available.

7 Limitations

While our research provides valuable insights, it is important to recognize its limitations. Specifically, our dataset was constructed with a narrow focus on certain domains and PII entities. Although this

allowed us to develop a flexible methodology adaptable to various domains, it also limits the dataset's generalizability.

Due to the lack of suitable manually annotated data, we were unable to fully assess the pipeline's transferability to real-world data. Applying the pipeline to real-world scenarios may require adjusting the hyperparameters to better fit specific datasets.

Another significant limitation is that the generated PII entities only describe the text's author. In many cases, personal information about individuals closely related to the author could also be classified as PII, but such cases are not covered in our dataset.

While SPY enhances privacy-preserving technologies, we are aware that misuse of this dataset could lead to privacy violations, data manipulation, or exploitation of personal data in ways that harm individuals.

To mitigate these risks, we have taken several precautions. First, our dataset is entirely synthetic, ensuring that no real-world PII is exposed or used in its creation. Second, we emphasize the importance of deploying this dataset solely in privacy-enhancing contexts, such as the development of security systems, compliance tools, or research dedicated to improving data protection standards.

We acknowledge, however, that the nature of this work may still be subject to unethical uses, including malicious attempts to create models that de-identify PII in ways that exploit sensitive information. We strongly discourage any use of our dataset that aims to undermine privacy protections or misuse the generated synthetic data for harmful purposes.

8 Ethics Statement

All entities in the generated dataset are artificial, and should solely be utilized for the task of PII detection. The methodology for dataset generation is transparent and can be adapted to any domain and any quantity of PII entities.

References

Griffin Adams, Alexander Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From sparse to dense: Gpt-4 summarization with chain of density prompting. *Preprint*, arXiv:2309.04269.

AI@Meta. 2024. Llama 3 model card.

Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. 2023. Santacoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*.

Tuomas Aura, Thomas A. Kuhn, and Michael Roe. 2006. Scanning electronic documents for personally identifiable information. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, page 41–50, New York, NY, USA. Association for Computing Machinery.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024a. Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv preprint arXiv:2402.15343*.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024b. Nuner: Entity recognition encoder pre-training via llm-annotated data. *Preprint*, arXiv:2402.15343.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniele Faraglia. 2014. Faker. `https://github.com/joke2k/faker`.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Alistair EW 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*, pages 214–221.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Holmes Langdon, Crossley Scott, Baffour Perpetual, King Jules, Burleigh Lauryn, Demkin Maggie, Holbrook Ryan, Reade Walter, and Howard Addison. 2024. The learning agency lab - pii data detection.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Simona Mazzarino, Andrea Minieri, and Luca Gilli. 2023. Nerpii: A python library to perform named entity recognition and generate personal identifiable information. In *Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023)*.

Microsoft. 2021. Presidio. `https://microsoft.github.io/presidio/`.

Arvind Narayanan and Vitaly Shmatikov. 2010. Myths and fallacies of" personally identifiable information". *Communications of the ACM*, 53(6):24–26.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining?

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *Preprint*, arXiv:2311.08526.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. Universalner: Targeted distillation from large language models for open named entity recognition. *Preprint*, arXiv:2308.03279.

A DeBERTa hyperparameters

Hyperparameters	**Value**
Optimizer	AdamW
Adam β_1, β_2	0.9, 0.999
Adam ϵ	1e-6
Warm-up step	100
Epochs	2
Batch size	1
Grad. accumulation steps	4
Context size	1800
Learning rate (LR)	5e-6

Table 6: `DebertaV3-base` hyperparameters

B Data Analysis

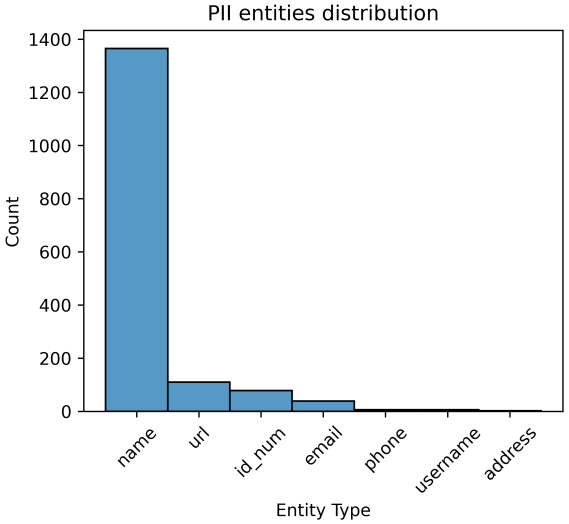

Figure 2: The distribution of entities present in the Kaggle PII dataset illustrates its highly imbalanced nature.

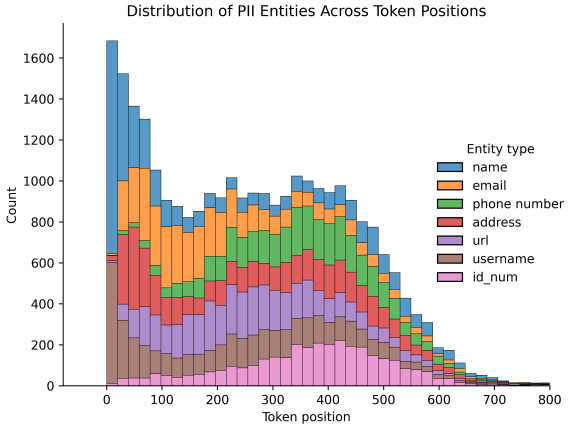

Figure 3: Distribution of PII entities in the Legal Questions Domain of the SPY Dataset. The x-axis shows token positions up to 800, though a small number of entities extend up to 1600 tokens. Tokenization was performed using spaCy's English tokenizer.

C UniNER "NAME" Class Prediction

UniNER "NAME" Class Prediction Example
Hi all, I 'm Nuria Batista, reaching out because I 'm in a bit of a tricky situation and I 'm hoping someone with legal expertise can offer some guidance. I 'm a marketing coordinator at an advertising agency, and one of our clients is accusing us of breach of contract. My team and I have reviewed the contract thoroughly, and we 're confident that we 've met all of the requirements. However, the client is still pushing for a refund and is threatening to take legal action against me, specifically at the office of attorney Emily Brown, located at 123 Main St, San Francisco, CA 94105.

Table 7: Name Nuria Batista is correctly classified as PII, while Emily Brown is misclassified due to the fact that UniNER doen't differentiate between PII and non-PII.

D PII Dataset Gegenration Pipiline Prompts

Extract the following personal information entities from the provided text, ensuring that only personally identifiable information (PII) related to the author of the text is captured:

- **Person:** Names of the author. Do not include names of other people, famous authors, celebrities, or historical figures.
- **Email:** Personal email addresses of the author.
- **Phone:** Personal phone numbers of the author.
- **ID:** Personal identification numbers of the author (e.g., Social Security Number, passport number).
- **URL:** URLs that are personal to the author and lead to pages containing personal data (e.g., the author's personal blogs, social media profiles).
- **Username:** Personal usernames of the author for online platforms.
- **Address:** Personal home addresses of the author.

Text: "text"

Format your response in JSON as follows:
{{ "person": ["list of the author's personal names"],
"email": ["list of the author's personal emails"],
"phone": ["list of the author's personal phone numbers"],
"id": ["list of the author's personal IDs"],
"url": ["list of the author's personal URLs"],
"username": ["list of the author's personal usernames"],
"address": ["list of the author's personal addresses"]
}}

If there is no information for a particular category, return an empty list for that category.

Figure 4: LLaMA-3-70B prompt for extracting PII entities from text.

Generate a biography of a fictional man named ***<generated-name-goes-here>***.

Occupation: any average job you can come up with
Personality: describe in 5 sentences

Present results in json format with fields "occupation": str, "personality": str

Figure 5: Prompt for biography generation. Placeholder "<generated-name-goes-here>" is replaced with random name.

Step 1) Look through the personality of the text author and pretend to be that person.

occupation: ***<generated-occupation>***
personality: ***<generated-personality>***

Step 2) Use the following instructions to generate a text:

<domain-specific-instructions>

Requirements:
- At any circumstance do not include any personal information in generated text.

Respond only with generated text with no commentary. Here goes your text:

Figure 6: Prompt for generating texts, which do not contain any personal information. Placeholders "<generated-*>" and "<domain-specific-instructions>" are replaced with according descriptions.

Text: {}

Task: You are an author of the above Text. Your task is to add new placeholders in the Text from the list below. You will be penalized for mentioning any placeholders other than what is listed below!

Here is the list of placeholders representing your personal information:
<author_personal_name> - A full or partial name of the text author
<author_personal_email> - An author's email address
<author_personal_username> - An author's username on any website, social media etc.
<author_personal_phone_number> - A phone number associated with the author or his relatives
<author_personal_url> - A link to author's social media page or personal website
<author_personal_address> - A full or partial street address that is associated with the author, such as home address
<author_personal_identifier> - A number or sequence of characters that could be used to identify an author, such as a social security number or medical policy number

Requirements:
- Do NOT change existing placeholders
- Distribute placeholders evenly throughout your text, do not stack them all in one place
- New text must be more entity-dense than the previous one

Respond only with updated text with no commentary. Here goes an updated text:

Figure 7: Prompt for adding PII placeholders into the text.

Text: {}

Task: You are given a Text, which contains author's personal information. Your task is to add new entities, which are not related to the text author. Generate entities using the following classes: name, email, username, phone number, url, address, identifier.

Requirements:
- At any circumstance DO NOT change author's personal information in the above text
- Newly generated entities should not disclose the personal information of the author of the text

Respond only with updated text with no commentary. Here goes an updated text:

Figure 8: Prompt for adding entities with personal information that are not relatted to text author.

Extract the following personal information entities from the provided text, ensuring that only personally identifiable information (PII) related to the author of the text is captured:

- **Person:** Names of the author. Do not include names of other people, famous authors, celebrities, or historical figures.
- **Email:** Personal email addresses of the author.
- **Phone:** Personal phone numbers of the author.
- **ID:** Personal identification numbers of the author (e.g., Social Security Number, passport number).
- **URL:** URLs that are personal to the author and lead to pages containing personal data (e.g., the author's personal blogs, social media profiles).
- **Username:** Personal usernames of the author for online platforms.
- **Address:** Personal home addresses of the author.

Text: {text}

Format your response in JSON as follows:
{ "person": ["list of personal names"], "email": ["list of personal emails"], "phone": ["list of personal phone numbers"], "id": ["list of personal IDs"], "url": ["list of personal URLs"], "username": ["list of personal usernames"], "address": ["list of personal addresses"] }

If there is no information for a particular category, return an empty list for that category.

Figure 9: Prompt for extracting PII from text.

E Generation examples

I am Juriņa, currently working on a project for a local museum, where we are about to launch a new exhibit featuring artifacts from an ancient civilization. As part of our agreement with the exhibit's benefactors, we are required to provide a guarantee that the artifacts will be displayed and handled with the utmost care and respect. However, during the installation process, one of our team members accidentally damaged one of the artifacts, which has left us in a bit of a predicament.

We've tried to repair the damage as best we could in-house, but we're not confident that our repairs meet the benefactors' standards, and we're worried that if the benefactors find out, they may withdraw their support for the exhibit. I've consulted with our in-house legal team at @JessicaL82.com, who have advised us to document everything and prepare for the worst-case scenario. However, I was wondering if anyone with more experience in this area could offer some guidance on the best course of action to take. Specifically, I'd like to know if we have any legal obligations to disclose the damage to the benefactors, and what our options are for addressing the situation without compromising the exhibit's integrity or the museum's reputation.

You can reach me at morgan_berg@taylor.com or call me at 001-758-439-5672x685 for any additional information or questions. I'd be eternally grateful for any guidance or insight that could help us navigate this tricky situation. Please feel free to share your thoughts and opinions at https://facebook.com/GreyHound2001. I'd appreciate any advice from experienced professionals like Dr. Sophia Patel, sophia.patel@lawfirm.com, 415-456-7890, or Mr. Michael Brown, mbrown@consultant.com, 202-555-1234, with a track record of handling sensitive issues similar to ours. You can also reach out to me directly at 9497 Samuel Walk New Claudia, MH 32542, and I will make sure to respond promptly.

Figure 10: Example of a text generated in a legal domain. NER and PII entities shown in color.